

# Universidad de Valladolid

## Escuela de Ingeniería Informática


TRABAJO FIN DE GRADO

**Grado en Ingeniería Informática**
**Mención Tecnologías de la Información**

# Integración de políticas de privacidad de aplicaciones web


Autor:
**D. Javier Miguel Tejero Álvarez**
Tutores:
**Dña. Mercedes Martínez González**
**D. Amador Aparicio de la Fuente**

# Resumen

El objetivo de este proyecto consiste en obtener información estructurada de distintas fuentes de datos heterogéneas sobre políticas de privacidad y ofrecer un acceso unificado a los usuarios para su consulta.

Para ello se construye un almacén de datos, un Warehouse, describiendo todo el proceso para su elaboración, desde la obtención de datos hasta su implementación final donde se facilitará a los usuarios su posterior consulta a través de una aplicación web.

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

## 1.1. Motivación

Todos los días a lo largo de nuestra vida, utilizamos servicios y aplicaciones que nos producen beneficios y motivaciones tras su uso, como puede ser comunicarnos con otras personas, intercambio y compartición de información a través de la red.

El problema que la totalidad de los servicios presenta, ya sea proporcionándolo esos beneficios a través de su uso de forma gratuita o de pago, es que nuestros datos e información puede ser recogida por la organización que suministra dicho servicio con fines particulares sin que los usuarios seamos conscientes de ello. Cuando se realiza una recogida y tratamiento de nuestros datos, es necesario la declaración de una política de privacidad.

Una política de privacidad [1] es un documento que utiliza una organización para revelar cómo recopilan, analizan, comparten y protegen la información personal de los usuarios.

Las jurisdicciones legales de todo el mundo requieren que las organizaciones pongan sus políticas de privacidad a disposición de sus usuarios, y leyes como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea [2] y la Ley de Protección de Privacidad en Línea de California (CalOPPA) en los Estados Unidos [3] establecen expectativas específicas sobre políticas de privacidad.

Un problema derivado de esta especificación de las políticas de privacidad es que muchos usuarios de Internet que utilizan estos servicios no comprenden las políticas de privacidad. Hay estudios [4] que muestran que los usuarios no hacen el esfuerzo de leer estos avisos de privacidad para determinar el tratamiento de la información porque percibe que consumen demasiado tiempo, son demasiado complicados de comprender, extensos y confusos.

Para abordar este problema, hay organizaciones que desde hace mucho tiempo trabaja en la creación de corpus de políticas de privacidad a escala web que realice un análisis de todos estos documentos de privacidad y proporcione información descriptiva acerca del servicio. Hay muchos enfoques para el desarrollo de estos corpus basados en la realización de un análisis manual para posteriormente proporcionar un aprendizaje automático de verificación de la integridad, a través de la categorización de la política de privacidad.

La integración de datos [5] es la combinación de procesos técnicos y de negocio utilizados para combinar datos de orígenes dispares en una única información valiosa y con significado. Una solución completa de integración de datos proporciona datos fiables procedentes de diversos orígenes para dar soporte a un canal de distribución de datos preparado para datos empresariales.

En la figura 1.1, se puede ver un esquema general de integración de forma muy representativa de donde se encuentran las fuentes de datos, los agentes y el repositorio de donde se guardan los metadatos o se reformulan las consultas para acceder a las fuentes de datos.

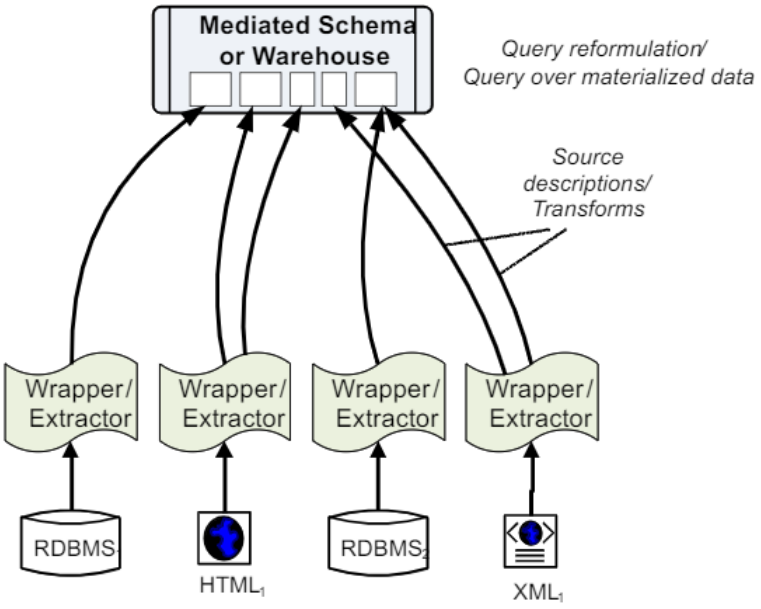


Figura 1.1: Esquema general de integración de información. Fuente: [6]

De este modo, podemos extraer grandes volúmenes de datos de varios orígenes, transformarlos en cualquier estilo y cargarlos en el almacén de datos. Los estilos de integración de datos adicionales incluyen la réplica de datos, utilizando la sincronización o la distribución de datos prácticamente en tiempo real. Por otro lado, también tenemos la virtualización de datos con el acceso a datos abstractos desde varios orígenes mediante la creación de una vista virtual sin almacenamiento para los usuarios que necesitan acceder y consultar datos bajo demanda.

## 1.2. Objetivos

El objetivo principal de este proyecto es construir un sistema de integración de políticas de privacidad.

Debido a que las distintas fuentes de datos heterogéneas almacenan distinto tipo de información estructurada y actualizada, necesitaremos analizar y diferenciar claramente cuál es la información estructurada que realmente necesitamos, que reside en una pero no en otra, analizar y comparar entre fuentes.

Además de reunir la distinta información en una sola, permitirá proporcionar un repositorio a los usuarios para mejorar la búsqueda de información relativa a su uso, tratamiento, exploración de datos, aclaraciones y clasificación de ese servicio en cuanto a los beneficios y perjuicios de su utilización.

## 1.3. Entorno

El entorno del proyecto estará formado por todo lo que concierne a políticas de privacidad y se compone de dos páginas web como fuentes de datos, donde cada una dispone de su interfaz de usuario en la que se presentan los datos. Las otras dos fuentes que se usarán será mediante repositorios descargables en local. Para que se produzca dicha presentación de los datos a los usuarios una vez realizada la integración, se usará una interfaz de usuario, que incluya todas las incorporaciones derivadas del análisis del documento de la política de privacidad para cada uno de los servicios seleccionados.

Debido a la gran cantidad de servicios que hay en las fuentes de datos, este análisis se realizará en las áreas de aplicación de los servicios de correo electrónico y servicios de mensajería muy utilizados en el día a día, tales como: Gmail, Hotmail, Yahoo!, WhatsApp, Telegram, Discord, Element, Session, Signal, Messenger, iCloud, ProtonMail, Slack y Twitter.

Se usarán estas cuatro fuentes de datos para obtener los datos que se utilizarán para la realización de un repositorio centralizado con toda la información estructurada y disponible.

## 1.4. Estructura de la memoria

El presente documento de memoria está estructurado en capítulos. En primer lugar, el presente capítulo, con una introducción del trabajo que se ha estado desarrollando, cuáles eran las motivaciones y objetivos del proyecto.

El segundo capítulo se compone de la planificación y método de trabajo para el desarrollo del proyecto, junto con un análisis de los riesgos que nos podemos encontrar.

En el tercer capítulo, se realiza el conocimiento de las fuentes de datos, utilizadas para la construcción del repositorio central integrador.

En el cuarto capítulo se describe cuáles son las tecnologías utilizadas en el proyecto y explicaciones de decisiones tomadas en la parte del integrador.

En el quinto capítulo se detalla el análisis, desde el punto de vista de toda la parte que nutre al almacén central como de la aplicación de usuario, donde puede visualizar la información. Se detallan los requisitos funcionales, no funcionales, de información y casos de uso.

En el sexto capítulo se especifica todo lo que tiene que ver con el diseño, es decir, arquitecturas del sistema.

El capítulo séptimo es la implementación de las dos soluciones que se llevarán a cabo, construcción del Warehouse y la aplicación de consulta.

En el octavo capítulo se detallan las pruebas realizadas para comprobar que los sistemas se han creado correctamente y funciona tal y como se quiere dar uso de él.

El último capítulo se presentan conclusiones obtenidas del trabajo realizado y el posible trabajo futuro.

Finalmente, los anexos, donde está incluido las pruebas del proceso de carga, copia de seguridad y actualización, el manual de usuario y el contenido de la documentación adicional proporcionada.

# Capítulo 2

# Planificación

A la hora de elaborar un proyecto, tenemos que seguir una planificación para desarrollar las tareas teniendo en cuenta los espacios temporales para el análisis de la información, materiales, diseño e implementación del integrador para conformar el nuevo repositorio y su posterior aplicación para consultar. Además del tiempo, debemos tener en cuenta el orden en el cual se llevarán a cabo todas estas tareas con las precedencias.

Otra de las cuestiones que debemos tener en cuenta son los riesgos con los que nos vamos a enfrentar, que provocarán retrasos y será necesario reestructurar esta planificación.

## 2.1. Planificación inicial

La metodología empleada en su elaboración será un proceso incremental. Tenemos cinco fases claramente diferenciadas, en las cuales cada una se subdivide para obtener un nivel de granularidad más fino. La primera fase es la del conocimiento de las fuentes de datos y obtención de conocimientos de las diferentes formas de integración, aprendiendo la utilización de RDF en paralelo. En la siguiente fase se realiza un análisis donde se especifican casos de uso y funcionalidad necesaria. A continuación, el diseño lógico y físico, seguido de la construcción y puesta en marcha del integrador a través de una interfaz en la implementación. Finalmente, en la última fase, se realizan las pruebas y la depuración del software realizado.

En las figuras 2.1 y 2.2, se puede obtener una vista general de las tareas programadas para comenzar y finalizar antes de una determinada fecha, con el diagrama de Gantt.

## 2.2. Análisis de riesgos

La gestión de riesgos en cualquier proyecto informático que se esté elaborando es uno de los aspectos más importantes en el desarrollo de este. Hay que identificarlos, analizarlos, valorarlos y tomar una serie de decisiones para evitar que ocurran (plan de protección). Del mismo modo, si se ha producido ese riesgo, es necesario una serie de medidas que intenten evitar, en la medida de lo posible, el daño causado y que este sea el menor posible (plan de contingencia).

La lista de riesgos identificados y analizados iría de las tablas 2.1 a 2.9.

DIAGRAMA DE GANTT

| | Modo de | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
|---|---|---|---|---|---|---|
| 1 | | **Conocer las fuentes de datos.** | **16 días** | **vie 05/02/21** | **vie 26/02/21** | |
| 2 | | Analisis de la documentación, información y materiales disponibles para cada una de las fuentes de datos. | 11 días | vie 05/02/21 | vie 19/02/21 | |
| 3 | | Abstracción de cada fuente de datos, con la obtención de esquemas relacionales. | 2 días | lun 22/02/21 | mar 23/02/21 | 2 |
| 4 | | Scraping web y práctica en JSON. | 3 días | mié 24/02/21 | vie 26/02/21 | 3 |
| 5 | | RDF y lenguaje de consulta de RDF (SPARQL). | 15 días | vie 05/02/21 | jue 25/02/21 | |
| 6 | | | | | | |
| 7 | | **Análisis.** | **19 días** | **lun 01/03/21** | **jue 25/03/21** | **1** |
| 8 | | Análisis del Data Warehouse. | 9 días | lun 01/03/21 | jue 11/03/21 | |
| 9 | | Análisis de la aplicación web. | 10 días | vie 12/03/21 | jue 25/03/21 | 8 |
| 10 | | | | | | |
| 11 | | **Diseño.** | **23 días** | **vie 26/03/21** | **mar 27/04/21** | **7** |
| 12 | | Diseño del Data Warehouse. | 11 días | vie 26/03/21 | vie 09/04/21 | |
| 13 | | Diseño de la aplicación web. | 12 días | lun 12/04/21 | mar 27/04/21 | 12 |
| 14 | | | | | | |
| 15 | | **Implementación.** | **26 días** | **mié 28/04/21** | **mié 02/06/21** | **11** |
| 16 | | Implementación del Data Warehouse. | 13 días | mié 28/04/21 | vie 14/05/21 | |
| 17 | | Implementación de la aplicación web. | 13 días | lun 17/05/21 | mié 02/06/21 | 16 |
| 18 | | | | | | |
| 19 | | **Pruebas.** | **9 días** | **jue 03/06/21** | **mar 15/06/21** | **17** |
| 20 | | Depuración. | 5 días | jue 03/06/21 | mié 09/06/21 | |
| 21 | | Validación. | 4 días | jue 10/06/21 | mar 15/06/21 | 20 |
| 22 | | | | | | |
| 23 | | Realización de la memoria | 90 días | vie 12/02/21 | jue 17/06/21 | |

Figura 2.1: Planificación inicial de tareas.

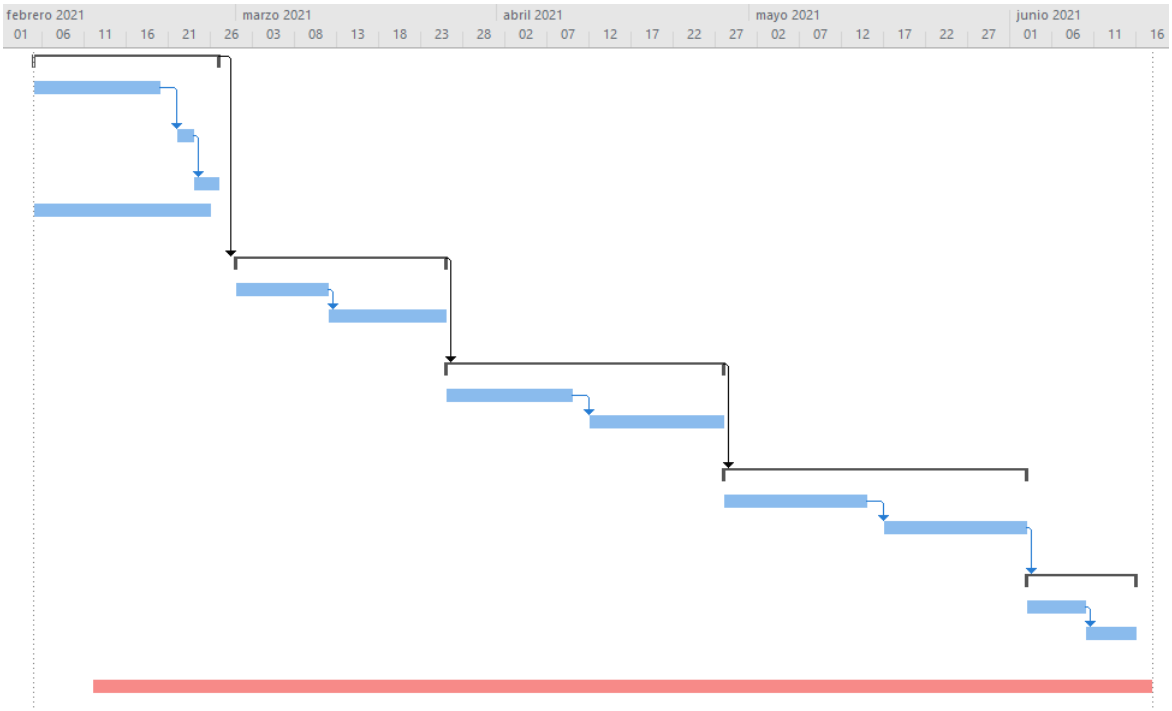


Figura 2.2: Diagrama de Gantt de planificación inicial temporal.

| **Riesgo 01** | Falta de experiencia en las tecnologías empleadas. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Media. |
| Indicador | Uso de tecnologías desconocidas y retraso temporal en la planificación. |
| Plan de protección | Formación para aprender acerca de esa tecnología empleada. |
| Plan de contingencia | Buscar ayuda externa para orientar el proyecto correctamente. |

Tabla 2.1: Riesgo 01: Falta de experiencia en las tecnologías empleadas.

| **Riesgo 02** | Enfermedad del alumno. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Baja. |
| Indicador | Ninguno, de difícil predicción. |
| Plan de protección | Establecer una planificación hitos e ir adelantado a la finalización de esos hitos. |
| Plan de contingencia | Replanificar tareas a realizar en función del tiempo que no ha estado disponible, según lo planificado. |

Tabla 2.2: Riesgo 02: Enfermedad del alumno.

| **Riesgo 03** | Modificación del proyecto. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Media. |
| Indicador | Cambio en los requisitos iniciales del proyecto. |
| Plan de protección | Reuniones tras la finalización de fases, entregas parciales de comprobación. |
| Plan de contingencia | Identificar las partes afectadas para incorporar los nuevos cambios lo antes posible. |

Tabla 2.3: Riesgo 03: Modificación del proyecto.

| **Riesgo 04** | Pérdida de información. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Media. |
| Indicador | Fallos en algún repositorio intermedio o final, sin datos disponibles. |
| Plan de protección | Copias de seguridad. Realizarla lo más próximo a cualquier pérdida. |
| Plan de contingencia | Recuperación de la información desde la última copia de seguridad útil realizada. |

Tabla 2.4: Riesgo 04: Pérdida de información.

| **Riesgo 05** | Corrupción de la información. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Bajo. |
| Indicador | La información no se corresponde con la original. |
| Plan de protección | Realización de copias de seguridad. |
| Plan de contingencia | Recuperación de la información desde la última copia de seguridad útil realizada. |

Tabla 2.5: Riesgo 05: Corrupción de la información.

| **Riesgo 06** | Equipo de trabajo estropeado. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Bajo. |
| Indicador | No se enciende, apagados y bloqueos espontáneos. |
| Plan de protección | Realización de copias de seguridad. Usar almacenamiento externo con replicación. |
| Plan de contingencia | Recuperación de la información desde la última copia de seguridad útil realizada o dispositivo de almacenamiento. |

Tabla 2.6: Riesgo 06: Equipo de trabajo estropeado.

| **Riesgo 07** | Fuentes de datos no disponibles. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Bajo. |
| Indicador | Servidores caídos donde no se puede acceder a los datos para su descarga. |
| Plan de protección | Copias de seguridad de los datos. |
| Plan de contingencia | Recuperación de la información desde la última copia de seguridad útil realizada. |

Tabla 2.7: Riesgo 07: Fuentes de datos no disponibles.

| **Riesgo 08** | Actualización |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Media. |
| Indicador | Organizaciones con incorporación de nuevos análisis de documentos de políticas. |
| Plan de protección | Copias de ficheros y comparativa de los nuevos. |
| Plan de contingencia | Identificación de los cambios e incorporarlos. |

Tabla 2.8: Riesgo 08: Actualización.

| **Riesgo 09** | Pruebas y validación escasas. |
|---|---|
| Impacto | Crítico. |
| Probabilidad | Bajo. |
| Indicador | Consultas sin resultados o resultados erróneos. |
| Plan de protección | Establecimiento de consultas de entrada y los resultados esperados. |
| Plan de contingencia | Identificar y solucionar los problemas lo antes posible. |

Tabla 2.9: Riesgo 09: Pruebas y validación escasas.

## 2.3. Planificación final

Una vez enumerados los riesgos a los que se ha sometido el proyecto y posibles retrasos que podíamos tener a medida que se avanzaba, la nueva planificación de tareas no ha variado mucho desde la planificación inicial.

La fecha de finalización del proyecto ha cambiado para aplicar las modificaciones. Si bien, cualquier retraso que se ha tenido principalmente en la parte inicial con el conocimiento de las fuentes de datos y obtención de conocimientos del proyecto, se han visto compensados en las fases posteriores de análisis y diseño.

El nuevo diagrama de Gantt con la planificación de tareas finalmente ha sido el de las figuras 2.3 y 2.4.

DIAGRAMA DE GANTT

| | Modo de | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
|---|---|---|---|---|---|---|
| 1 | | **Conocer las fuentes de datos.** | **20 días** | **vie 05/02/21** | **jue 04/03/21** | |
| 2 | | Analisis de la documentación, información y materiales disponibles para cada una de las fuentes de datos. | 15 días | vie 05/02/21 | jue 25/02/21 | |
| 3 | | Abstracción de cada fuente de datos, con la obtención de esquemas relacionales. | 2 días | vie 26/02/21 | lun 01/03/21 | 2 |
| 4 | | Scraping web y práctica en JSON. | 3 días | mar 02/03/21 | jue 04/03/21 | 3 |
| 5 | | RDF y lenguaje de consulta de RDF (SPARQL). | 15 días | vie 05/02/21 | jue 25/02/21 | |
| 6 | | | | | | |
| 7 | | **Análisis.** | **17 días** | **vie 05/03/21** | **lun 29/03/21** | **1** |
| 8 | | Análisis del Data Warehouse. | 7 días | vie 05/03/21 | lun 15/03/21 | |
| 9 | | Análisis de la aplicación web. | 10 días | mar 16/03/21 | lun 29/03/21 | 8 |
| 10 | | | | | | |
| 11 | | **Diseño.** | **22 días** | **mar 30/03/21** | **mié 28/04/21** | **7** |
| 12 | | Diseño del Data Warehouse. | 10 días | mar 30/03/21 | lun 12/04/21 | |
| 13 | | Diseño de la aplicación web. | 12 días | mar 13/04/21 | mié 28/04/21 | 12 |
| 14 | | | | | | |
| 15 | | **Implementación.** | **25 días** | **jue 29/04/21** | **mié 02/06/21** | **11** |
| 16 | | Implementación del Data Warehouse. | 12 días | jue 29/04/21 | vie 14/05/21 | |
| 17 | | Implementación de la aplicación web. | 13 días | lun 17/05/21 | mié 02/06/21 | 16 |
| 18 | | | | | | |
| 19 | | **Pruebas.** | **9 días** | **jue 03/06/21** | **mar 15/06/21** | **17** |
| 20 | | Depuración. | 5 días | jue 03/06/21 | mié 09/06/21 | |
| 21 | | Validación. | 4 días | jue 10/06/21 | mar 15/06/21 | 20 |
| 22 | | | | | | |
| 23 | | Realización de la memoria | 99 días | vie 05/02/21 | mié 23/06/21 | |

Figura 2.3: Planificación final de tareas.

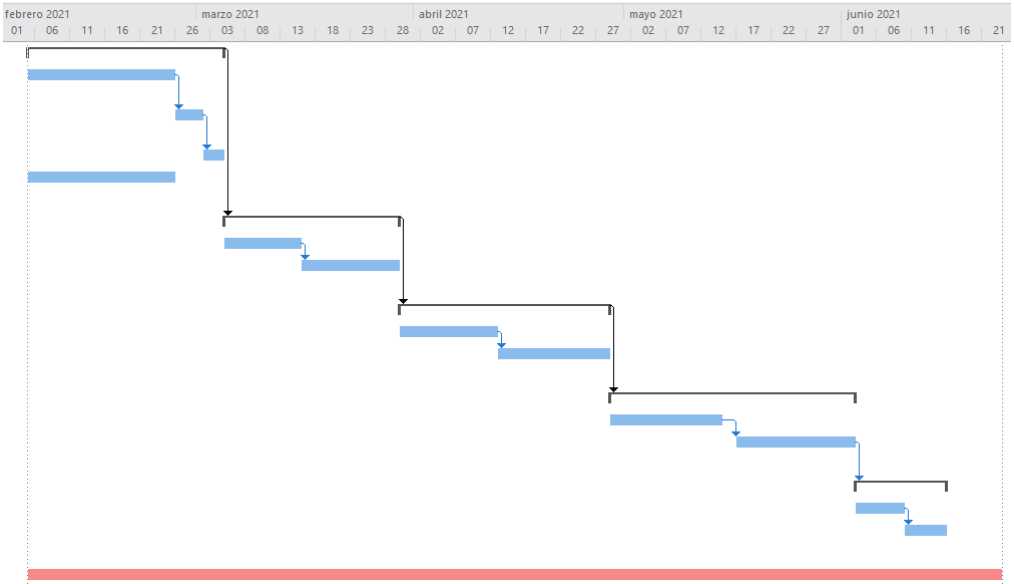


Figura 2.4: Diagrama de Gantt de planificación final temporal.

# Capítulo 3

# Fuentes de datos

Anteriormente ya hemos hablado de la importancia que tiene la privacidad en las personas. Cuando se utilizan distintos servicios, que necesitan nuestros datos y las comunicaciones se realizan a través de ellos, es necesario tener un conocimiento acerca de ellos. Entre todos los servicios disponibles, nos centraremos en aquellos servicios de correo y mensajería.

Hay organizaciones que se encargan de realizar análisis sobre la privacidad de los servicios. Las cuatro fuentes de datos con las que se van a trabajar tienen el mismo objetivo común, la simplificación de la información presentada haciéndola más descriptiva y categorizada, pero con prácticas totalmente diferentes.

## 3.1. Estructura

Antes de comenzar con la descripción y la obtención de la información que está disponible en cada fuente de datos sobre políticas de privacidad, veremos cómo está estructurado un documento de una política de privacidad [7]. Toda política de privacidad incluye los siguientes aspectos:

- Información del responsable del tratamiento o de su representante: Suministrar información del responsable del tratamiento de los datos o de su represente.
- Datos del responsable en materia de protección de datos: Deben aparecer los datos de contacto del responsable del tratamiento de los datos personales. Así, se indicará el nombre o razón social y NIF, las direcciones electrónicas y postales y el número de teléfono del responsable de tratamiento o su representante.
- Información sobre la finalidad del tratamiento: Se informará sobre la finalidad del tratamiento, es decir, para qué serán usados los datos personales que se recojan (por ejemplo, fines estadísticos o para el envío de información). También se debe especificar la legitimación jurídica con la que se han obtenido los datos. Es necesario contar con el consentimiento expreso de los usuarios para registrar y tratar sus datos personales.
- Terceros destinatarios de los datos personales: Informar a los usuarios de los terceros destinatarios a quienes se envían los datos personales, si esta cesión de datos se produce, informando de quiénes son esos encargados de tratamiento.

  También es necesario informar de las cookies de terceros, puesto que estas registran datos personales de los usuarios. Es decir, identificar también a estos terceros destinatarios como

otros encargados de tratamiento, que podrán tener acceso a los datos personales del usuario, si este da su consentimiento para ello.

- Transferencias internacionales: Si se van a transferir datos fuera de países de la Unión Europa, se tiene que informar de ello en la política de privacidad.

- Plazo de conservación de los datos: Indicar el plazo máximo durante el que se conservarán los datos personales facilitados por los usuarios.

- Derechos ARCO: Información referente a cómo pueden ejercer los usuarios sus derechos ARCO (Acceso, Rectificación, Cancelación y Oposición) y a través de que canal hacerlo.

- Explicación sobre el uso de decisiones individuales automatizadas: Si se realizan decisiones en el tratamiento automatizado de datos, como son la elaboración de perfiles de usuario, es necesario una explicación de la lógica de su fundamento. Se explican los efectos y el alcance que el proceso automatizado tiene sobre él.

## 3.2. PrivaSeer

PrivaSeer [1] es un corpus de casi un millón y medio de políticas de privacidad de sitios web, realizado a través de una evaluación automática por aprendizaje de las tecnologías de seguimiento utilizadas, regulaciones y órganos de regulación que utilizan para velar por la aplicación efectiva de su código.

Para obtener todo este conjunto de corpus de políticas de privacidad de los distintos servicios, se llevan a cabo una canalización importante de pasos que se describen:

- Recopilación los documentos.

  - Uso de *Common Crawl*: Proporcionar una instantánea de la web al incluir los nuevos rastreos de dominios populares y nuevos.

  - Selección de URL: El paradigma de la privacidad es el de "Aviso y elección". Hay una tendencia a usar palabras clave en las URL como pueden ser "Política de privacidad" y "Protección de datos".

  - Rastreo web: Rastreo de las URL seleccionadas mediante la utilización de programas como *Scrapy*, donde se eliminan aquellas que conducen a páginas de error, duplicados y sin respuesta.

- Filtrado de documentos.

  - Detección de idioma: Identificación del idioma de inglés frente a otros, usando el paquete de Python *Langid* y descartar aquellos documentos con un idioma distinto del inglés.

- Clasificación de documentos: Se realiza un etiquetado manual para establecer un hito del trabajo realizado para posteriormente realizar un aprendizaje automático no supervisado usando distintos algoritmos.

  También se hace uso de un aprendizaje automático supervisado usando las características extraídas de la URL, otro de la página web y otro usando características de ambos.

- Verificación cruzada de URL: Comparación de las URL obtenidas hasta el momento, con las que están disponibles en la página de inicio, normalmente en el pie de página de su sitio web para cada organización.

- Extracción de contenido: Eliminación del contenido estándar en todas las políticas de privacidad dejando únicamente el texto de la política de privacidad.

- Detección de duplicados y casi duplicados: Eliminación de los elementos duplicados exactos aplicando un hash a todos los documentos sin procesar y descartando las copias de hash exactas.

  Entradas similares producen valores hash similares, por lo que calculando la distancia entre los valores obtenidos dentro de un mismo dominio se pueden localizar estos elementos casi duplicados.

- Análisis de Corpus.

  - Legibilidad: Muy importante para la toma de decisiones de lectura o ignoración del documento de la política de privacidad por parte de los usuarios.

  - Modelado de temas: Usando un aprendizaje automático no supervisado, que extrae la distribución más probable de palabras en temas, a través de un proceso iterativo. Identificar diferentes categorías de análisis de datos en las políticas de privacidad y la extracción de temas comunes para los párrafos.

  - Extracción de frases clave: Las palabras clave y frases clave se prestan bien para resumir el contenido de una política de privacidad.

  - Similitud de las políticas de privacidad web: Algunas políticas de privacidad tenían exactamente la misma redacción en varias secciones, y solo se diferenciaban por el nombre de la organización, siendo prácticamente similares.

Como resultado, tenemos el corpus a través de una interfaz de usuario y accesible por cualquier persona, usando un motor de búsqueda para el descubrimiento.

### 3.2.1. Abstracción

En PrivaSeer, como en cualquier otra fuente, toda la información que se presenta al usuario a través de su interfaz tiene que ser alojada en algún almacén físico. Algunas fuentes sí que proporcionan algún mecanismo para que sea accesible por cualquier persona a esta información

ya sea bien a través de su API o descargables con los datos. Al realizar la abstracción, estaríamos determinando cómo estarían alojados estos datos como si fuese una base de datos relacional.

El esquema relacional resultante sería:

SERVICIO(nombre_servicio, fuente, fecha_version)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- fuente: Enlace al documento actual de la política de privacidad del servicio analizado. No se corresponde con el documento realmente analizado. Formato del dato: Texto.
- fecha_version: Fecha relativa a la versión del documento de privacidad en la cual se ha realizado el análisis para la incorporación en la fuente de datos. Puede diferir del que hay actualmente en la fuente. Formato del dato: Fecha.

INDUSTRIA(nombre_servicio, nombre_industria, subnombre_industria)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- nombre_industria: Actividad donde se desarrolla el servicio. Formato del dato: Texto.
- subnombre_industria: Subactividad donde tiene desarrollo el servicio. Formato del dato: Texto.

TRACKING(nombre_servicio, categoria_track)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- categoria_track: Ámbito de la tecnología de seguimiento de la información por el servicio [terceros, cookies, registros, baliza web, huellas dactilares, cookies flash, identificación del dispositivo, identificación de publicidad]. Formato del dato: Texto.

REGULACION(nombre_servicio, regulador)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- regulador: Valor de la regulación y acuerdos del tratamiento y circulación de la información [GDPR, COPPA, Privacy Shield, CalOPPA]. Formato del dato: Texto.

ORGANO(nombre_servicio, organo_regulador)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.

- organo_regulador: Valor del órgano de supervisión y autorregulación de la información [NAI, DAA, EDAA]. Formato del dato: Texto.

### 3.2.2. Formato de archivo y disponibilidad de los datos

Todos los datos descritos en la abstracción se encuentran accesibles a través de una interfaz web. Se extraen ficheros en formato de texto JSON (JavaScript Object Notation) tras realizar un *web scraping* con herramientas para extraer información de su sitio web donde están presente los datos.

Para obtener una respuesta, es necesario realizar una búsqueda en un cuadro de texto por el nombre del servicio o su URL, seleccionándolo a través de un botón de tipo radio. Una vez realizada la búsqueda, podemos ordenar por coincidencia, por ranking, por legibilidad, por categorías y por niveles.

La fecha de incorporación de los datos es del año 2019 y la frecuencia de actualización de los datos presentes es baja.

### 3.2.3. Completitud de la fuente

En cuanto al análisis realizado de políticas de privacidad tiene lugar en distintos sectores bien diferenciados, cuyas categorías y subcategorías están bien jerarquizadas y diferenciadas donde nos encontramos:

- **Consumidor y cadena de suministro:** Minorista, industria textil y moda, diseño, bienes de consumo, servicios al consumidor, transporte / camión / ferrocarril, al por mayor, instalaciones de servicios, muebles, material y equipo comercial, producción alimentaria, logística y cadena de suministro, servicios individuales y familiares, cosméticos, artículos de lujo y joyería, artes y oficios, marítimo, envases y contenedores, textiles, agricultura, plásticos, comercio internacional y desarrollo, almacenamiento, entrega de paquetes / fletes, lácteos, supermercados, pesca, tabaco y ganadería.

- **Tecnología de la información y electrónica:** Servicios y tecnología de la información, software de computadora, internet, telecomunicaciones, electrónica de consumo, servicios de información, seguridad informática y de redes, hardware de computadora, redes informáticas, semiconductores, inalámbrico, desarrollo de programas y nanotecnología.

- **Médico:** Salud, bienestar y fitness, atención sanitaria y hospitalaria, práctica médica, dispositivos médicos, productos farmacéuticos, biotecnología, cuidado de la salud mental, veterinario, medicina alternativa.

- **Deportes, medios y entretenimiento:** Entretenimiento, servicios para eventos, deportes, producción de medios, imprenta, medios en línea, música, artículos deportivos, relaciones públicas y comunicaciones, medios de difusión, fotografía, diseño gráfico, juegos de computadora, instalaciones y servicios recreativos, periódicos, artes escénicas, bellas artes, películas y animación.

- **Civil, mecánica y eléctrica:** Construcción, automotor, fabricación eléctrica / electrónica, ingeniería industrial o mecánica, maquinaria, petróleo y energía, materiales de construcción, energía renovable y medio ambiente, arquitectura y planificación, productos químicos,

automatización industrial, utilidades, aerolíneas / aviación, minería y metales, ingeniería civil, aviación y aeroespacial, productos de papel y forestales, vidrio, cerámica y hormigón, construcción naval y fabricación de ferrocarriles.

- **Sin ánimo de lucro:** Gestión de organizaciones sin ánimo de lucro, servicios medioambientales, filantropía, recaudación de fondos, seguridad pública y gestión de organizaciones sin ánimo de lucro.

- **Educación:** Gestión educativa, entrenamiento y coaching profesionales, editorial, aprendizaje electrónico, educación superior, investigación, redacción y edición, educación primaria / secundaria, investigación de mercado, museos e instituciones, traducción y localización, laboratorio de ideas y bibliotecas.

- **Gobierno, defensa y legal:** Práctica de la abogacía, servicios legales, seguridad e investigaciones, organización cívica y social, administración gubernamental, instituciones religiosas, oficina ejecutiva, defensa y espacio, políticas públicas, organización política, relaciones gubernamentales, asuntos internacionales, aplicación de la ley, militar, resolución alternativa de litigios, poder judicial y oficina legislativa.

- **Viajes, comida y hostelería:** Hostelería, ocio, viajes y turismo, alimentos y bebidas, restaurantes, vinos y bebidas espirituosas, apuestas y casinos.

- **Finanzas, marketing y recursos humanos:** Marketing y publicidad, servicios financieros, inmobiliaria, seguros, consultoría de gestión, dotación de personal y contratación, contabilidad, recursos humanos, gestión de inversiones, banca, bienes de raíces comerciales, capital riesgo y capital privado, subcontratación / deslocalización, importación y exportación, banca de inversión y mercados de capitales.

En las áreas de aplicación en la que se va a desarrollar la integración de servicios de correo y mensajería, tenemos los siguientes servicios disponibles en la fuente de datos: Gmail, Hotmail, Yahoo!, WhatsApp, Telegram, Discord, Messenger, iCloud, ProtonMail, Slack y Twitter.

## 3.3. Terms of Service; Didn't Read

Otra de las fuentes de datos es Terms of Service; Didn't Read o en su nombre abreviado ToS;DR [8]. Se inspira de la sigla TL;DR que significa en inglés "Too Long; Didn't Read" (demasiado largo, sin leer). Este término es utilizado habitualmente cuando un bloque de texto es demasiado largo y los usuarios dejan de leerlo.

Como hemos mencionado anteriormente, los términos y condiciones que acostumbramos a ver suelen ser demasiado largos, pero no por eso dejan de ser importantes ya que los derechos digitales dependen de ellos. Por ello, a través de una elaboración propia manual, quieren que las clasificaciones realizadas informen acerca de los derechos que se dispone.

En el año 2011 tiene lugar este movimiento para crear aplicaciones web que otorguen a los usuarios un control sobre sus datos valiosos y vida privada, categorizando el seguimiento realizado y clasificando la importancia, tanto positiva como negativa, que tiene ese seguimiento con referencias al texto. Comienza un análisis legal donde un equipo de trabajo se encarga de realizar análisis y contribuciones en público, que pueden ser debatidas por el resto. Todo el software es gratuito con datos abiertos al público general y de manera transparente.

Cualquier usuario puede aportar su valoración respecto a una serie de casos estándares y clasificarlo en el propio texto, la cual posteriormente es validada dicha aportación.

### 3.3.1. Abstracción

En ToS;DR, a diferencia de lo que ocurría con PrivaSeer, la información se presenta al usuario a través de su interfaz de usuario y también es accesible por su API, que tiene que ser alojada en algún almacén físico. Cuando realizamos la abstracción de la fuente de datos, estaríamos determinando cómo estarían alojados estos datos como si fuese una base de datos relacional.

El esquema relacional resultante sería, junto con la explicación de cada uno de los atributos asociados a la información que guarda:

SERVICIO(nombre_servicio, fuente, clasificacion)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- fuente: Enlace al documento actual de la política de privacidad del servicio analizado. No se corresponde con el documento realmente analizado. Formato del dato: Texto.
- clasificacion: Calificación a través de un índice [A-E] del trato justo, respeto y abuso de información por parte del servicio. Formato del dato: Texto.

DATOS(nombre_servicio, caso_track, titulo, referencia_texto, descripcion_tldr, valoracion, puntuacion)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- caso_track: Clasificación del seguimiento realizado del servicio en el documento de la política. Formato del dato: Texto.
- titulo: Especificación resumida detallada del *caso_track* de la política. Formato del dato: Texto.
- referencia_texto: Referencia a la parte del texto del documento de la política de privacidad para ese *caso_track*. Formato del dato: Texto.
- descripcion_tldr: Información adicional de la referencia al texto en ese *caso_track*. Formato del dato: Texto.
- valoracion: Asignación de la valoración cualitativa [bueno, malo, neutro] para este *caso_track*. Formato del dato: Texto.
- puntuacion: Asignación de un valor cuantitativo [0-100] en esa *valoracion* y *caso_track*. Formato del dato: Numérico.

### 3.3.2. Formato de archivo y disponibilidad de los datos

Los datos se presentan mediante ficheros en formato de texto JSON (JavaScript Object Notation) por medio de su API y accesibles también a través de descargables de un repositorio de GitHub. En dicha API se encuentran descritos todos los servicios sobre los cuales han realizado el análisis por parte de los usuarios y contribuyentes. También adicionalmente dispone de una interfaz de usuario para presentar los datos de forma que sea más accesible al usuario.

Para obtener una respuesta a través de dicha interfaz, es necesario realizar una búsqueda en un cuadro de texto por el nombre del servicio. Una vez realizada la búsqueda, obtenemos los servicios que coinciden con lo introducido y podremos visualizar el que buscamos seleccionándolo. Si por el contrario hacemos uso de la API, tenemos que localizar el fichero que se corresponde con el servicio.

La fecha de inicio de incorporación de los datos es del año 2011 y la frecuencia de actualización de los datos presentes es mensualmente.

### 3.3.3. Completitud de la fuente

En cuanto al análisis realizado de políticas de privacidad, tiene lugar en distintas áreas, sin una clasificación clara de las áreas. Los servicios se pueden incluir en alguna de las categorías generales como son:

- Servicios de comercio electrónico.
- Servicios de mensajería.
- Servicios tecnológicos.
- Servicios bancarios.
- Servicios de redes sociales.
- Servicios informativos y televisivos.
- Servicios de restauración.
- Servicios de videojuegos.
- Servicios educativos.
- Servicios de salud.
- Servicios inmobiliarios.

El análisis realizado para cada servicio nos proporciona las categorías de seguimiento referenciando las distintas partes del documento de la política de privacidad, una descripción del caso, puntuaciones y valoraciones detalladas acerca de la información que utiliza.

En las áreas de aplicación en la que se va a desarrollar la integración de servicios de correo y mensajería, tenemos los siguientes servicios disponibles en la fuente de datos: Gmail, Hotmail, Yahoo!, WhatsApp, Telegram, Discord, Element, Session, Signal, Messenger, iCloud, ProtonMail, Slack y Twitter.

## 3.4. OPP-115 Manual

La siguiente fuente de datos que se ha utilizado para obtener información de políticas de privacidad para los servicios seleccionados ha sido OPP-115 (Online Privacy Policies, set of 115) Manual que proporciona *Usable Privacy* [9]. Se encarga del desarrollo y resolución de los problemas de privacidad mediante un proyecto de política de privacidad utilizable, realizado sobre los 115 servicios más populares y utilizados.

Tiene como origen en que las políticas de privacidad del lenguaje natural se han convertido en el estándar de facto para abordar las expectativas de "notificación y elección" en la web. Sin embargo, como los usuarios generalmente no leen estas políticas y son aquellos a quienes les cuesta entenderlas, se tiene como objetivo abordar este problema mediante el desarrollo de formatos legibles por máquina para transmitir las prácticas de datos de un sitio web o servicio.

En *Usable Privacy* se basan en los avances recientes en el procesamiento del lenguaje natural (NLP), el modelado de preferencias de privacidad, la colaboración abierta de las tareas (*crowdsourcing*) y el diseño de la interfaz de privacidad para desarrollar un marco práctico basado en la política de privacidad de lenguaje natural existente de un sitio web. Esto va a permitir a los usuarios controlar de manera más significativa su privacidad al poder realizar búsquedas de un servicio y determinar las distintas categorías en los que se realiza seguimiento, referenciando dicho análisis al texto junto con una clasificación del nivel de lectura, sin requerir la cooperación adicional de los operadores del sitio web.

Para obtener el corpus de políticas de privacidad de los distintos servicios, se llevan a cabo una serie de pasos realizados por expertos en derecho que consisten en:

- Selección de la política de privacidad: Selección de los servicios a analizar por medio de una preselección del sitio web basado en la relevancia y un submuestreo basado en el sector de aquellos sitios web de nivel superior de *DMOZ.org*.
- Proceso y esquema de anotación: Identificación de diferentes categorías de práctica de datos (seguimiento) y sus atributos descriptivos de las políticas de privacidad. Está dividida en diez grandes categorías:

    1. Recopilación, uso de primera parte: Cómo y por qué un proveedor de servicios recopila información del usuario.
    2. Recopilación de terceros, uso compartido: Cómo la información del usuario puede ser compartida o recopilada por terceros.
    3. Elección, control del usuario: Opciones de control disponibles para los usuarios.
    4. Acceso de usuario, edición y eliminación: Cómo los usuarios pueden acceder, editar o eliminar su información.
    5. Retención de datos: Cuánto tiempo se almacena la información del usuario.
    6. Seguridad de los datos: Cómo se protege la información del usuario.
    7. Cambio de política: Cómo se informará a los usuarios sobre cambios en la política de privacidad.

8. No rastreo: Cómo se respetan las señales para el rastreo y la publicidad en línea.
9. Audiencias internacionales y específicas: Prácticas que pertenecen solo a un grupo específico de usuarios.
10. Otros: Etiquetas secundarias adicionales para texto introductorio o general, información de contacto y prácticas no cubiertas por las otras categorías. Normalmente no tienen referencias al usuario.

- Contenido de la política: Cada política de privacidad es leída por tres expertos de forma independiente, en la que realizaron una categorización en base al esquema de anotación anterior.

  Cualquier parte del texto está cubierto por una de estas diez categorías. La recopilación para el uso en primera parte y de terceros es de las preocupaciones principales.

- Consolidación del trabajo de los anotadores: Una vez realizado el análisis por separado por parte de los expertos, se realiza la combinación de prácticas de datos etiquetadas, si esas prácticas se refieren a la misma práctica subyacente expresada por el texto. Entonces, esa práctica se consolida si todos pertenecen a la misma categoría. Para que esa parte del segmento sea clasificada, debe tener al menos mayoría por parte de los expertos. Se establece un sistema de puntuación: alta para dos prácticas que están asociadas con el mismo texto de la política, mientras que se asigna una puntuación baja en caso contrario.

- Sitio web de exploración de datos: En el sitio web se incorpora la estructura y utilidad del conjunto de datos para visualizar las anotaciones de prácticas de datos con los textos de las políticas de privacidad. Proporciona una comparación de las políticas en base a su estructura, categorías, nivel de lectura.

Si bien, seguir todos estos pasos de una manera manual por parte de los anotadores puede que no sea una solución práctica para todos los documentos de políticas que hay para los servicios y sus respectivas actualizaciones con el paso del tiempo. Por ello, se plantea el hecho de que se automatice parcialmente esta clasificación de segmentos de texto en etiquetas.

Se emplea para ello las similitudes semánticas entre palabras en el vocabulario de las políticas de privacidad ya realizadas, reconociendo que el vocabulario en este dominio es especializado, pero no completamente estandarizado. La única diferencia existente es que da lugar a un desarrollo de método no supervisado.

### 3.4.1. Abstracción

En OPP-115 Manual, la información se presenta al usuario a través de una interfaz de usuario y un repositorio descargable con toda la información analizada, que tiene que ser alojada en algún almacén físico. Realizamos la abstracción de la fuente de datos, cómo estos datos estarían alojados si fuese una base de datos relacional.

El esquema relacional resultante sería, junto con la explicación de cada uno de los atributos asociados a la información que guarda:

SERVICIO(nombre_servicio, fuente, fecha_version, nivel_lectura)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- fuente: Enlace al documento actual de la política de privacidad del servicio analizado. No se corresponde con el documento realmente analizado. Formato del dato: Texto.
- fecha_version: Fecha relativa a la versión del documento de privacidad sobre el cual se ha realizado el análisis. Formato del dato: Fecha.
- nivel_lectura: Nivel de legibilidad de la política de privacidad a través de un nivel de grado de Estados Unidos (Flesch–Kincaid Grade Level). Formato del dato: Texto.

PRIMERAPARTE(nombre_servicio, tipo_informacion, proposito, accion, modo_coleccion, tipo_coleccion, anonimizacion, eleccion_usuario, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- tipo_informacion: Tipo de información que recolecta la organización. Formato del dato: Texto.
- proposito: Cuál es el objetivo por el cual recopila o utiliza esa información. Formato del dato: Texto.
- accion: Especifica si realmente la política hace o no hace algo explícitamente. Formato del dato: Texto.
- modo_coleccion: Si la colección es implícita (sin conocimiento del usuario) o explicita (con conocimiento del usuario) o no especifica. Formato del dato: Texto.
- tipo_coleccion: Cómo recopila, rastrea u obtiene la información del usuario. Formato del dato: Texto.
- anonimizacion: Indica si la práctica de información o datos está vinculada a la identidad del usuario o si es anónima. Formato del dato: Texto.
- eleccion_usuario: Indica si se ofrecen explícitamente opciones de usuario para esta práctica. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

TERCEROS(nombre_servicio, tipo_informacion, proposito, terceros, accion, tipo_coleccion, anonimizacion, eleccion_usuario, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- tipo_informacion: Tipo de información que recolecta la organización tercera. Formato del dato: Texto.
- proposito: Cuál es el objetivo por el cual un tercero recopila o utiliza esa información. Formato del dato: Texto.

- terceros: Cuál es el tercero involucrado en la práctica de datos. Formato del dato: Texto.
- accion: Especifica si realmente la política hace o no hace algo explícitamente. Formato del dato: Texto.
- tipo_coleccion: Cómo recopila, rastrea u obtiene el tercero la información del usuario. Formato del dato: Texto.
- anonimizacion: Indica si la práctica de información o datos está vinculada a la identidad del usuario o si es anónima. Formato del dato: Texto.
- eleccion_usuario: Indica si se ofrecen explícitamente opciones de usuario para esta práctica. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

ELECCION(nombre_servicio, tipo_eleccion, alcance_eleccion, tipo_informacion, proposito, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- tipo_eleccion: Tipo de elección del usuario u opciones de control de privacidad disponibles al usuario. Formato del dato: Texto.
- alcance_eleccion: Alcance de la elección o control del usuario de la recopilación y uso en primera parte o en terceros. Formato del dato: Texto.
- tipo_informacion: El tipo de información que aplica la elección del usuario. Formato del dato: Texto.
- proposito: Objetivo que se aplica la elección del usuario. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

ACCESO(nombre_servicio, derechos_acceso, alcance_acceso, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- derechos_acceso: Opciones que se ofrecen a los usuarios para acceder, editar y eliminar la información que la organización del servicio tiene sobre ellos. Formato del dato: Texto.
- alcance_acceso: Si se ofrece el acceso a la información, a qué información se aplica. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

RETENCION(nombre_servicio, periodo_retencion, proposito_retencion, tipo_informacion, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- periodo_retencion: Cuanto tiempo se almacenan los datos. Formato del dato: Texto.
- proposito_retencion: Propósito que se aplica a la práctica de retención. Formato del dato: Texto.
- tipo_informacion: El tipo de información para el que se especifica el periodo de retención. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

SEGURIDAD(nombre_servicio, medida_seguridad, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- medida_seguridad: El tipo de seguridad que implementa el servicio para proteger la información de los usuarios. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

CAMBIO(nombre_servicio, tipo_notificacion, tipo_cambio, opciones_usuario, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- tipo_notificacion: Cómo se notifica al usuario cuándo cambia la política de privacidad. Formato del dato: Texto.
- tipo_cambio: Tipo de cambios en la política que se notifica a los usuarios. Formato del dato: Texto.
- opciones_usuario: Opciones que se ofrecen al usuario cuando cambia la política. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

AUDIENCIA(nombre_servicio, grupo_audiencia, referencia_texto)

- nombre_servicio: Nombre único del servicio que se describe. Formato del dato: Texto.
- grupo_audiencia: A que audiencia se refiere el segmento de la política. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia. Formato del dato: Texto.

### 3.4.2. Formato de archivo y disponibilidad de los datos

Los datos están presentes de dos formas. A través de un repositorio descargable de ficheros en formato CSV (valores separados por comas), donde uno de sus campos incorpora notación en formato de texto JSON (JavaScript Object Notation). La otra es a través de una interfaz web de usuario en la cual se puede visualizar mejor las distintas categorías de seguimiento y tratamiento de la información en las que se clasifica el texto de la política. Para obtener una respuesta es necesario realizar una búsqueda del fichero CSV en el repositorio que vamos a analizar y posteriormente obtener la columna de ese archivo que contiene el formato JSON con los datos.

La fecha de incorporación de los datos es del año 2015 y la frecuencia de actualización de los datos presentes es muy baja, casi nula.

### 3.4.3. Completitud de la fuente

En cuanto al análisis realizado manualmente de las 115 políticas de privacidad, se obtiene el conjunto a través de las Tendencias de Google (Google Trends) para obtener políticas de privacidad de un conjunto más diverso de sitios web [10]. Cada una de las tendencias se analiza de diferente forma para obtener los resultados.

También se realiza un muestreo por categorías, donde se organiza el conjunto de datos de acuerdo con las categorías principales de Alexa / DMOZ / ODP (listado y categorización de enlaces a páginas web) en Estados Unidos para obtener un conjunto de datos sobre el que hacer comparaciones relativas entre sitios web de la misma categoría. En DMOZ / ODP la mayoría de los sitios web ya están categorizados.

Las categorías donde que se utilizan serían:

- Servicios de letras.
- Servicios de juegos.
- Servicios de juventud y adolescencia.
- Servicios de referencia.
- Servicios de mensajería.
- Servicios comerciales.
- Servicios de negocio.
- Servicios de salud.
- Servicios informativos.
- Servicios regionales (Estados Unidos).
- Servicios de sociedad.
- Servicios de ordenadores.

- Servicios del hogar.
- Servicios recreativos.
- Servicios de ciencias.
- Servicios informáticos.
- Servicios deportivos.

En el análisis realizado destacamos que para cada servicio nos proporciona las distintas categorías con un mayor nivel de detalle de seguimiento que cualquier otra fuente de datos, referenciando las distintas partes del documento de la política de privacidad donde tiene lugar y una calificación del nivel de lectura de la política.

En las áreas de aplicación en la que se va a desarrollar la integración de servicios de correo y mensajería tenemos, dentro de los 115 documentos analizados, los siguientes servicios disponibles en la fuente de datos: Gmail, Hotmail y Yahoo!.

## 3.5. OPP-115 Automática

La última fuente de datos que se ha utilizado para obtener información de políticas de privacidad para los servicios seleccionados ha sido OPP-115 (Online Privacy Policies, set of 115) Automática que proporciona *Usable Privacy* [9]. Se encarga del desarrollo y resolución de los problemas de privacidad mediante un proyecto de política de privacidad utilizable en la cual para cada una de las categorías de seguimiento que pueden hacer la organización al prestar el servicio, presenta los datos de forma que sea sencillo de determinar y filtrar para los usuarios.

Es una versión ampliada de la fuente de datos anterior (OPP-115 Manual), que surge para resolver el problema de la limitación del número de servicios con sus políticas de privacidad analizadas. En esta nueva versión, se amplían a 7000 servicios, donde el propósito y finalidad por el que realizan el desarrollo es el mismo que para OPP-115 Manual, pero con una granularidad más gruesa, menor nivel de detalle en la clasificación de las referencias al texto. La única diferencia es que utilizan mecanismos automatizados basado en el análisis realizado para OPP-115 Manual.

La categorización del *tracking* realizado es la misma y únicamente diferenciándose de los algoritmos utilizados y que no es realizado por expertos

### 3.5.1. Abstracción

A diferencia de lo que ocurre en la fuente de OPP-115 Manual, la información se presenta al usuario únicamente a través de una interfaz de usuario y que nuevamente esa información de la interfaz tiene que ser alojada en algún almacén físico. Por ello, realizamos la abstracción para determinar cómo estarían alojados estos datos como si fuese una base de datos relacional.

Con la información que nos proporciona, tenemos el esquema relacional resultante, junto con la explicación de cada uno de los atributos asociados a la información que almacena:

SERVICIO(nombre_servicio, fuente, fecha_version, nivel_lectura)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- fuente: Enlace al documento actual de la política de privacidad del servicio analizado. No se corresponde con el documento realmente analizado. Formato del dato: Texto.
- fecha_version: Fecha relativa a la versión del documento de privacidad sobre el cual se ha realizado el análisis. Formato del dato: Fecha.
- nivel_lectura: Nivel de legibilidad de la política de privacidad a través de un nivel de grado de Estados Unidos (Flesch–Kincaid Grade Level). Formato del dato: Texto.

DATOS(nombre_servicio, categoria_track, referencia_texto)

- nombre_servicio: Nombre único del servicio de la política de privacidad que se describe. Formato del dato: Texto.
- categoria_track: Categorización del texto de la política de privacidad. Ámbito de la tecnología de seguimiento de la información por el servicio. Formato del dato: Texto.
- referencia_texto: Texto resultante clasificado al que se hace referencia y describe para una *categoria_track* de seguimiento. Formato del dato: Texto.

### 3.5.2. Formato de archivo y disponibilidad de los datos

Los datos se encuentran accesibles a través de una interfaz web. Se extraen ficheros en formato de texto JSON (JavaScript Object Notation) tras realizar un *web scraping* con herramientas para extraer información de su sitio web donde están presente los datos.

Para obtener una respuesta, es necesario realizar una búsqueda en un cuadro de texto por el nombre del servicio. Una vez realizada la búsqueda, se elige una de las coincidencias que ha encontrado la interfaz web a esa búsqueda del servicio, si es que está disponible. Nos proporciona el nombre del servicio y la URL para comprobar que realmente esa coincidencia de texto es la que estamos buscando.

La fecha de incorporación de los datos es del año 2017 y la frecuencia de actualización de los datos presentes es muy baja.

### 3.5.3. Completitud de la fuente

En cuanto al análisis realizado de políticas de privacidad, nuevamente tiene lugar en distintas áreas, sin una clasificación clara de las áreas. Incluyen desde:

- Servicios de comercio electrónico.
- Servicios de mensajería.
- Servicios bancarios.

- Servicios de redes sociales.
- Servicios informativos y televisivos.
- Servicios de restauración.
- Servicios de videojuegos.
- Servicios educativos.
- Servicios de salud.
- Servicios inmobiliarios.

El análisis realizado destacamos que para cada servicio nos proporciona las distintas categorías, con un nivel de detalle de seguimiento referenciando las distintas partes del documento de la política de privacidad donde tiene lugar y una calificación del nivel de lectura de la política.

En las áreas de aplicación en la que se va a desarrollar la integración de servicios de correo y mensajería, tenemos los siguientes servicios disponibles en la fuente de datos: Gmail, Hotmail, Yahoo!, WhatsApp, Telegram, Discord, Messenger, iCloud, ProtonMail, Slack yTwitter.

## 3.6. Valoración general y comparativa entre fuentes

Una vez realizado el conocimiento en profundidad, con su descripción y desarrollo, nos damos cuenta de que cada una de ellas es similar en cuanto al propósito de su implementación, pero presentan diferencias en cuanto al grado de detalle, refinamiento y servicios analizados. Tenemos tres grandes categorías en las que poder realizar la comparación:

- **Actualización y frecuencia de las fuentes:** La fuente que tiene un mayor nivel de actualización es ToS;DR, ya que constantemente están revisando las políticas de privacidad e incorporando nuevos casos de seguimiento conforme a las actualizaciones, evaluando y clasificando. Del mismo modo amplían incorporando análisis de nuevos servicios a esta fuente. Dejan un historial temporal visible de todos los casos introducidos.

  Por otro lado, la fuente con menor grado de actualización es OPP-115 Manual del año 2015, y que posteriormente se ha utilizado como base para el estudio automatizado de los 7000 servicios que tiene OPP-115 Automática, la cual desde que se elaboró no se ha realizado ninguna actualización.

  Con respecto a los servicios de correo y de mensajería que analizamos, las últimas modificaciones de algunos de sus documentos de políticas de privacidad han tenido lugar en:

  - Gmail: Febrero 2021.
  - Hotmail: Marzo 2021.
  - Yahoo!: Septiembre 2020.

- WhatsApp: Enero 2021.
- Telegram: Marzo 2021.
- Discord: Junio 2020.
- Wire: Mayo 2018.
- Matrix: Agosto 2020.
- Signal: Mayo 2018.

La fecha de actualización de estas políticas está totalmente desacoplada de la de las fuentes de datos donde se almacenan. La mayoría de las actualizaciones de sus documentos son del presente año o anterior. De aquí podemos sacar conclusiones para la toma de decisiones del tipo de almacén de datos que se va a utilizar a la hora de integrar, junto con una política de actualizaciones.

En principio, estas actualizaciones de las fuentes de datos serán baja o nula en cuanto a la variación. En estos casos, el mecanismo utilizado sería el de un Warehouse que integre toda la información disponible, en detrimento del empleo de una integración virtual usando un mediador, teniendo presente las ventajas e inconvenientes de utilizar uno u otro.

- **Completitud:** Un análisis realizado sobre una gran variedad de categorías, clasificando cada servicio, será más completa que una que solo cubre un determinado área y categorías.

  Una fuente de datos que abarque un mayor número de categorías y en ellas esté incluidos más servicios, será mucho mejor para una posible escalabilidad que aquellas fuentes con servicios limitados. En nuestro caso, PrivaSeer es la que presenta una mayor categorización y un mayor número de servicios analizados por categoría. Mientras que OPP-115 Manual abarca gran variedad de categorías, pero con pocos servicios analizados en cada una de ellas.

  En la mayoría de los casos, aquellos análisis realizados por una persona especializada será mejor que aquel que utiliza mecanismos automatizados para llevarlo a cabo. Por ello, OPP-115 Manual realizado por expertos será la más apropiada ya que tiene una categorización del seguimiento con un mayor grado de detalle. Del mismo modo, también tenemos ToS;DR, que es realizada por personas.

  Por otro lado, tanto PrivaSeer como OPP-115 Automática utilizan mecanismos automatizados para obtener los resultados, partiendo de una base.

- **Autoría:** En la autoría, relacionado con el punto anterior, OPP-115 Manual emplea tres personas especializadas en derecho para realizar la lectura y filtrado del documento y su consecuente categorización del seguimiento. ToS;DR también emplea a las personas que quieran realizar una aportación para su estudio y clasificación. La parte negativa de esta última es que puede darse el caso de que estas personas puede que no tengan conocimientos acerca de ello, realizando aportaciones erróneas y confusas. Para resolver el problema, se realizan revisiones finales por parte del personal de administración de la fuente de información.

  En cuanto a PrivaSeer y OPP-115 Automática utilizan algoritmos automáticos con determinada base de análisis del lenguaje natural para llevar a cabo el estudio del documento de las políticas de privacidad de los servicios que se analizan.

En la tabla 3.1 se puede ver la tabla comparativa entre las fuentes de datos para los criterios de actualización (Baja, Media, Alta), completitud (Baja, Media, Alta) y autoría.

| | Actualización | Completitud | Autoría |
|---|---|---|---|
| PrivaSeer | Baja | Alta | Algoritmos |
| ToS;DR | Alta | Media | Personal |
| OPP-115 Manua1 | Baja | Baja | Personal especializado |
| OPP-115 Automática | Baja | Media | Algoritmos |

Tabla 3.1: Comparación de las fuentes de datos.

# Capítulo 4

# Tecnologías utilizadas

Una vez que ya se han descrito las fuentes de datos y la información que recogen por separado, necesitamos un mecanismo que se encargue de unificar toda esta información. En base a este análisis, tendríamos que determinar cuál sería nuestra arquitectura en uno de los tres métodos disponibles: integración virtual, *warehousing* o mixto.

Por un lado, en la integración virtual, la información reside en las propias fuentes de datos siempre actualizadas, controlados por la organización que da el soporte. Por ello, puede darse el caso de tener problemas en el acceso a la fuente y eficiencia en el procesamiento. Las consultas se realizan directamente sobre los datos originales.

Por el otro lado, en el *warehousing*, la información reside en un repositorio servidor donde el control ya no es de la organización que creó esa fuente de datos y esta información tiene que ir actualizándose periódicamente según se realicen cambios desde las fuentes. Las consultas y procesos de analítica de datos ahora son más eficientes ya que no tenemos el problema de acceso a la fuente para cada consulta.

También veremos cuáles han sido las tecnologías que se emplearán para mostrar toda esa información al usuario.

## 4.1. Warehouse

El concepto de Warehouse lo entendemos como un almacén, en general. William H. Inmon desarrolló almacén de datos como "*a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions. The data warehouse contains granular corporate data*" [11]. Especificándolo en el ámbito que estamos desarrollando, un Data Warehouse se puede definir como un almacén de datos que recoge toda la información que hay en cada una de las fuentes de políticas de privacidad. Sobre la información que hay recogida de carácter descriptivo, se genera nueva información en forma de resúmenes, contabilizaciones e incluso se llega a un nivel superior de detalle para poder ofrecer información orientada a la toma de decisiones en cualquier momento, sin necesidad de esperar largos periodos en la realización de informes tediosos que tienen más probabilidad de error y acumulan un nivel muy superior de imprecisiones [12].

Es evidente el carácter integrador que ofrece un Data Warehouse, no sólo desde el punto de vista de recopilar información de todas las fuentes de datos que se use en la integración final, sino desde el punto de vista de la calidad de la información. Un Data Warehouse puede ser

la mejor herramienta para conseguir información de calidad, dónde los datos que se muestren sean válidos, se encuentren correctamente formateados y tengan un propósito bien definido. En definitiva, convertir los repositorios de la información en conocimiento útil al servicio prestado.

Conviene dejar claro que el Data Warehouse es un almacén de datos, y que el resto de técnicas tanto para la construcción del mismo, como para su mantenimiento y explotación forman parte de los sistemas de Data Warehousing, el *Business Intelligence*. Podríamos identificar el Data Warehousing como una tecnología, cuyo propósito es reunir información de distintas fuentes y efectuar un proceso de implementación de un proyecto Data Warehouse.

Si bien una vez definido el concepto de Data Warehouse, conviene agrupar una serie de características que nos permitirán distinguir y comprender que es y cuál es su funcionamiento principal:

- Es un depósito de datos. Los datos son independientes de los sistemas operativos o de las aplicaciones existentes. Simplemente satisfacen ciertos requisitos.
- Es una forma de arquitectura de estructura de datos. Permite atender consultas para la toma de decisiones, ya que dota a los sistemas de explotación del Data Warehouse de agregaciones y desagregación de datos de forma interactiva.
- Con el Data Warehouse se realiza el análisis del problema en términos de dimensiones. Permite realizar un análisis de los datos disponibles y tomar decisiones. El Data Warehouse es orientado a sujetos.
- Incluye un proceso que integra datos provenientes de diversas fuentes, y de distintos formatos. Tiene la capacidad de integrar datos heterogéneos para conformar información homogénea y precisa, dónde el hecho de generar conocimiento sea más sencillo. No sólo usa datos heterogéneos en el origen en cuanto a tipo de tecnología o formato del dato, sino de ámbito, como pudieran ser bases de datos relacionales.
- Es no volátil. La información que contiene un Data Warehouse se modifica con menor frecuencia y se puede considerar como "en tiempo no real", con actualización periódica en base a alguna política de actualización.
- Un Data Warehouse es un sistema que contiene su propia base de datos. Es decir, se puede ver como un sistema aparte de los sistemas de las organizaciones donde se extrae la información.
- La construcción y desarrollo de un Data Warehouse exitoso requiere la integración de varios componentes de tecnología y la habilidad para hacerlos funcionar todos juntos. Además, debe tenerse muy claro el propósito por el que se creará el Data Warehouse y saber que requisitos debe cubrir toma un papel crucial en el desarrollo de este.
- La finalidad de un Data Warehouse consiste en ayudar al usuario final a obtener conocimientos ayudando a anticiparse.

Además del uso de Warehouse como almacén, otro mecanismo que puede utilizarse para la integración es un enfoque de integración virtual [13]. En él se emplea un mecanismo de mediador en el cual los datos permanecen exclusivamente en las fuentes de datos y se obtienen de las fuentes cuando se consulta el sistema. Este enfoque sería la solución más eficaz en sistemas en

tiempo real, con actualizaciones de la información constantemente. Cuando una consulta llega a este mediador por parte de un usuario, se divide en subconsultas que recopilan información de las fuentes y se compone toda la información recibida actualizada, mostrándose al usuario. Esto contrasta con el enfoque que ya hemos definido de Warehouse, donde los datos se extraen de las fuentes de datos antes del tiempo de consulta, se transforman y se cargan en el almacén. En el momento de la consulta, se accede a este almacén creado, pero no a las fuentes de datos originales.

## 4.2. ETL: Extract, Transform and Load

El proceso Extract, Transform and Load (Extracción, Transformación y carga) [12] es el encargado de llevar los datos del sistema origen al Data Warehouse en el formato deseado.

No solamente se trata de realizar los tres pasos de sus siglas. También tenemos la oportunidad de generar datos de calidad para el Data Warehouse y poder conseguir que los procesos se hagan de forma eficaz y eficiente, además de conseguir que realmente se almacene información en el Data Warehouse y no solo una copia formateada de los datos del origen. Su funcionamiento es similar al de un *pipeline*, componente que proporciona una implementación del patrón de integración de canalizaciones y filtros proporcionando una entrada y una salida en serie por cada operación.

Evidentemente el proceso de consolidación incorpora integración de datos, filtros, normalización y toda clase de técnicas de manipulación de datos. Los puntos identificados en un diagrama ETL son cuatro grupos diferenciados que se detallan a continuación.

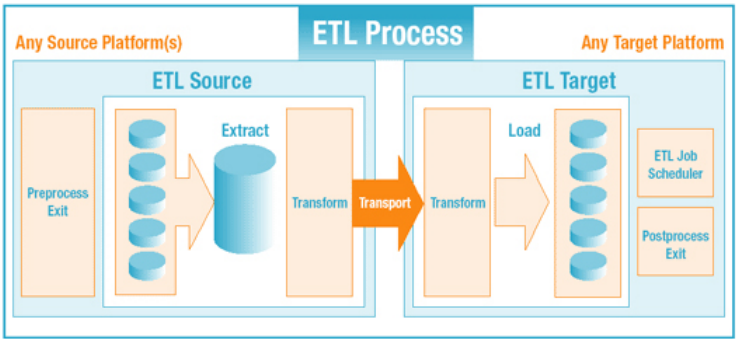


Figura 4.1: Proceso ETL. Fuente: [14]

### 4.2.1. Comunicación con las fuentes de datos: Interoperabilidad

El proceso de comunicación con las fuentes de datos se diferencia por motivos prácticos, es decir, en la teoría no se suelen contemplar técnicas o definir reglas de comunicación con las fuentes de datos porque directamente se suele ir al detalle de la tarea principal que queremos acometer, que es el proceso ETL de la figura 4.1. Pero se ha demostrado que para que el proceso ETL pueda comenzar sin inconvenientes y no conlleve retrasos innecesarios o sin contabilizar, se debe tener en cuenta una serie de cuestiones como son:

- ¿Dónde se encuentran las fuentes de datos?
- ¿Accedemos a copias de las fuentes de datos o a las fuentes de datos de producción?

- ¿Cuándo podemos acceder a las fuentes de datos y durante cuánto tiempo?
- ¿Qué restricciones de seguridad residen en las fuentes de datos para poder comenzar a extraer datos de ellas?

Una vez habiendo tenido especificadas las cuestiones anteriores y subsanadas, principalmente en el ámbito de seguridad e interoperabilidad con las fuentes de datos, podremos dar comienzo a la fase inicial del flujo ETL, la extracción.

Es muy importante prevenir todos los problemas de interoperabilidad y definirlos desde un inicio, ya que pueden ser problemas muy comunes que no se definen en las planificaciones de los proyectos, que lleven a que la resolución del proyecto global no termine en éxito.

En cuanto a los posibles requisitos de seguridad, suelen ser necesarias la aplicación de distintas configuraciones indicadas por la política de seguridad de las organizaciones que proporcionan estas fuentes.

### 4.2.2. Extracción

El proceso de extracción debe hacerse con un alto grado de estudio, ya que es el primer paso en el camino para poder obtener la información deseada.

Es importante conocer todas las estructuras de información que contienen las fuentes de datos para definir cómo recuperar la información necesaria. Por tanto, la definición de la fase de extracción no basta simplemente con lanzar consultas a las fuentes de datos para recuperar datos, sino que debe de hacerse para realizar el proceso de extracción de acuerdo con varias de las cuestiones definidas anteriormente.

Deberán de ejecutarse procesos de recuperación de información rápidos y que no tengan que repetirse por posibles fallos en la recuperación de la información, es decir, una vez que se va a la fuente del origen, debemos traernos todos los datos necesarios para a partir de aquí operar por cuenta propia en el Warehouse.

Se recupera la información necesaria y se carga toda esa información en el Data Warehouse, después de pasar por todo el proceso que veremos a continuación. Por tanto, el proceso ETL recupera los datos necesarios en el menor tiempo posible y directamente los formatea.

### 4.2.3. Transformación

El proceso de transformación puede verse como una caja negra, que recibe entradas de datos (inputs) y genera otros nuevos datos (outputs) que serán filtrados, transformados y estarán acorde con lo que se espera de ellos. Las fases serían:

- Obtención de datos: Se trata de obtener los datos que van a formar parte de la transformación. Generalmente los datos vienen procedentes de las fuentes. Deberá estipularse si la obtención de los datos se puede realizar en lote o bien se hace de forma unitaria. Sin esta tarea no puede realizarse el resto de la transformación.
- Tipo de transformación: En base al dato final que queramos obtener y el dato de entrada que tengamos, deberemos consultar los metadatos para que nos indiquen que tipo de transformación tenemos que realizar.

- Test de calidad de datos: Aplicar diferentes filtros de calidad a los datos. Actualizar registros con información libre de errores. Debemos tener presente las siguientes tres fases:

  1. Detección y definición de la tipología de errores.
  2. Búsqueda e identificación de los casos de error.
  3. Corrección de errores detectados.

- Marcar la transformación como finalizada: Una vez realizadas las tareas anteriores, se deberá marcar los datos como limpios y formateados para su posterior fase de carga, o como no válido para reportar de ello. Es importante conocer en qué puntos se pueden marcar los datos como corruptos para poder detectar errores en la fuente o simplemente para ver que se ha realizado de forma correcta el proceso de obtención de los datos.

### 4.2.4. Carga

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema destino. Este proceso puede abarcar una amplia variedad de acciones diferentes.

En algunas bases de datos se sobrescribe la información antigua con los datos nuevos. Normalmente, un Data Warehouse mantiene un histórico de los registros de manera que se pueda hacer una auditoría de estos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

Existen dos formas básicas de desarrollar el proceso de carga:

- Acumulación simple: La acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el periodo de tiempo seleccionado y transportar el resultado como una única transacción hacia el Data Warehouse.
- *Rolling*: El proceso de *rolling* por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones.

La fase de carga interactúa directamente con la base de datos del Warehouse. Al realizar esta operación se aplican todas las restricciones y disparadores que se hayan definido en ésta durante la fase de transformación (valores únicos, integridad referencial, campos obligatorios, rangos de valores).

Una dificultad adicional es asegurar que los datos que se cargan sean relativamente consistentes. Las múltiples fuentes de datos de origen tienen diferentes secuencias de actualización. En un sistema de ETL será necesario que se puedan detener ciertos datos hasta que todas las fuentes estén sincronizadas. Del mismo modo, cuando un Warehouse tiene que ser actualizado con los contenidos en un sistema de origen, es necesario establecer puntos de sincronización y de actualización, que veremos más adelante.

### 4.2.5. Riesgos de un Warehouse

Los procesos ETL pueden ser muy complejos. Un sistema ETL mal diseñado puede provocar importantes problemas operativos.

En un sistema operacional, el rango de valores de los datos o la calidad de éstos pueden no coincidir con las expectativas especificadas a la hora de determinar las reglas de validación o transformación.

Los Data Warehouse son alimentados desde distintas fuentes, que sirven a propósitos muy diferentes. El proceso ETL es clave para lograr que los datos extraídos de orígenes heterogéneos se integren finalmente en un entorno homogéneo.

La escalabilidad de un sistema de ETL durante su vida útil tiene que ser establecida durante el análisis de fuentes. Esto incluye la comprensión de los volúmenes de datos que tendrán que ser procesados.

#### 4.2.5.1. Frecuencia de actualización de los datos

En la descripción y toma de decisión de la implementación de un Warehouse que centralice e integre toda la información heterogénea de las distintas fuentes de datos, ya se mencionó que, aunque sea no volátil, es necesario disponer de un mecanismo de actualización. La desventaja con respecto a una integración virtual era el mantenimiento de los datos actualizados desde las fuentes de datos en todo momento a través de un repositorio central. En esa sección también se valoraron los riesgos de accesibilidad y las posibilidades de un enfoque u otro.

A la hora de llevar a cabo una actualización de este Warehouse, es necesario el diseño de una política de actualización. Uno de los problemas con que nos encontramos es que no existe un criterio genérico de actualización. Si bien, dar respuesta a preguntas como la frecuencia con la que se actualizan las fuentes usadas en las consultas o cómo y cuándo se realizará el filtrado de los nuevos datos en las fuentes pueden ayudarnos en su definición.

ToS;DR era la única fuente que presentaba una alta rotación de sus datos, en torno a un mes, por lo que será necesario una comprobación de que se ha añadido nueva información cada 25-30 días y aplicar los cambios. En las otras tres, la frecuencia de actualización es muy baja, casi inexistente, por lo que podemos unificar los criterios de ToS;DR y usarlo para el resto de fuentes.

Como para la realización de la integración se va a utilizar la herramienta SSIS, esta dispone de mecanismos de búsqueda y control del flujo. Por ello, si se establecen las condiciones adecuadas y los elementos de control de flujo descendente correctos, comprobando los elementos existentes para evitar duplicados y posibles actualizaciones de datos en alguna columna, no tendremos problemas.

Finalmente, reutilizando estas operaciones de ETL utilizadas para poblar el almacén, tendremos estos datos siempre actualizados en el Data Warehouse.

#### 4.2.5.2. Escalabilidad de las fuentes

Los sistemas de almacenamiento de datos facilitan el acceso a los datos integrados. Para lograr esto, en el Data Warehouse se ha tenido que procesar y modelar los datos en función de los requisitos funcionales. El mejor enfoque para desarrollar un almacén de datos es un proceso de desarrollo iterativo. Eso significa que la funcionalidad del almacén de datos, según lo solicitado

en los requisitos, se diseña, desarrolla, implementa y despliega en iteraciones (a veces llamado sprint o ciclo). En cada iteración, se agrega más funcionalidad al almacén de datos [15].

Sin embargo, al desarrollar el proyecto, incluso cuando se usa un enfoque iterativo, el esfuerzo y los costes asociados a él para agregar otra funcionalidad y cambios con nueva información generalmente aumenta debido a las dependencias existentes que deben ser atendidas.

El esfuerzo y coste de implementar inicialmente una primera información es relativamente bajo. Pero al escalar implementando la segunda y sucesivas, es necesario mantener la solución existente y ocuparse de las dependencias existentes de las fuentes de datos integradas previamente. Para asegurarse de que esta funcionalidad creada anteriormente no se interrumpa al implementar la nueva funcionalidad con los nuevos datos, se debe volver a probar todo el trabajo realizado anteriormente, al entrar en un estado de mantenimiento de todas las funciones. En muchos casos, la solución existente debe refactorizarse para mantener la funcionalidad.

La complejidad de los datos utilizados a la hora de la escalabilidad del Data Warehouse afecta en muchos factores, como son:

- Variedad de datos: El objetivo final del Warehouse es obtener información estructurada a partir de datos no estructurados o semiestructurados, para aumentar el valor proporcionado de los datos unificadamente.
- Volumen de datos: La tasa a la que las organizaciones generan y acumulan nuevos datos en las fuentes, aumentando con incorporación de nuevos servicios o ampliación de los existentes. Un mayor volumen de datos conduce a conjuntos de datos mucho más grandes.
- Velocidad de los datos: Aumenta rápidamente la velocidad a la que se crean los datos asociados a la variedad y volumen de los datos.
- Confiabilidad de los datos: Para tener confianza en los datos, se debe tener una sólida trazabilidad del origen de datos y una sólida integración de datos.

La escalabilidad de la cantidad de información disponible está relacionada con la actualización y adición de nueva información al almacén. En el caso de la escalabilidad del número de fuentes disponibles desde las cuales se pueden obtener los datos de la integración, así como la propia escalabilidad de una fuente con más información tiene prácticamente el mismo comportamiento que se ha descrito. Al introducir la nueva fuente que puede aportar más datos de los recogidos ya en el Data Warehouse, es necesario realizar la abstracción de esta fuente y visualizar que información que contiene es necesaria incorporar en él, modificando los esquemas. Todo el proceso ETL definido hasta el momento, que era especifico de la arquitectura que teníamos, será necesario modificarlo para incorporar estos nuevos cambios.

No llevar a cabo esta escalabilidad de las fuentes conduce a problemas de consolidación de los datos sin procesar donde suele ser requerido su acceso. Los cuatro problemas más destacados son:

- Acceso directo a los sistemas de origen: Los usuarios finales del sistema no deben acceder directamente a los datos de los orígenes. Esto expone datos sin procesar que a veces pueden ser privados o no transformados y permitir el acceso a estos datos podría eludir el acceso de seguridad.

- Datos brutos no integrados: Cuando se obtienen datos de varias fuentes de origen, los usuarios se quedan solos con la integración de datos brutos. Esto puede convertirse en una tarea tediosa y propensa a errores.

- Baja calidad de los datos: A menudo los datos de las fuentes de origen tienen problemas con la calidad de los datos. Antes de utilizar los datos, es necesario limpiarlos con la transformación para la obtención del Data Warehouse.

- Datos brutos no consolidados: Para analizar los datos de múltiples fuentes de origen, los datos a menudo requieren consolidación. Sin esta consolidación, los resultados del análisis de la organización no tendrán sentido.

#### 4.2.5.3. Seguridad

En la identificación de riesgos con los que nos íbamos a encontrar a la hora de desarrollar el proyecto de la sección 2.2, ya incluimos aspectos relacionados con la seguridad clasificando su impacto, probabilidad, indicador, plan de protección y plan de contingencia. La pérdida de información, corrupción de la información y la disponibilidad de las fuentes al hacer la integración pueden afectarnos en la seguridad.

La seguridad, como se establece en la Norma Internacional ISO / IEC 9126 [16], es uno de los componentes de la calidad del software. La seguridad de la información se puede definir como la preservación de la confidencialidad, integridad y disponibilidad de la información, en el que la confidencialidad garantiza que la información sea accesible solo para aquellos usuarios con privilegios de autorización. La integridad protege la precisión, la integridad de la información y los métodos de proceso. Finalmente, la disponibilidad garantiza que los usuarios autorizados tengan acceso a la información y los activos asociados cuando sea necesario. Por lo tanto, la seguridad del almacén de datos se define como los mecanismos que garantizan la confidencialidad, integridad y disponibilidad del almacén de datos y sus componentes.

Por un lado, para garantizar la protección de los datos de un Data Warehouse tradicional, el mecanismo más sencillo, conocido y a su vez eficaz es llevando a cabo *backups* de forma periódica. Si embargo, estos *backups* suelen fallar a la hora de incluir los datos más recientes. La mejor copia de seguridad que puede realizarse es la que se hace lo más cerca posible del punto de fallo del sistema ya que de esta forma las incoherencias en torno a los datos del *backup* y los reales serán mínimas.

En el otro lado, para conseguir una confidencialidad de los datos es necesario restringir y limitar los permisos y privilegios de los administradores del almacén, con controles de acceso por clave y controlar el lugar donde va a residir este almacén, si tendrá conexiones habilitadas desde Internet o será únicamente de almacenamiento local.

Se deben realizar pruebas automáticas de seguridad de forma periódica para comprobar si existe algún tipo de vulnerabilidad, un funcionamiento correcto, sin que esto afecte a la operativa del sistema.

## 4.3. SQL Server Integration Services

SQL Server Integration Services (SSIS) [17] es una plataforma para dar soluciones empresariales de transformaciones de datos e integración de datos. Es un servicio útil para resolver problemas complejos relacionados con la copia o descarga de archivos, la carga de almacenamientos de datos, la limpieza y minería de datos y la administración de datos y objetos de SQL Server.

Permite la extracción y transformación de datos de diversos orígenes como archivos de datos XML, CSV, JSON, archivos planos y orígenes de datos relacionales y después cargarlos en el destino.

En su catálogo incluye un amplio conjunto de tareas y transformaciones integradas, herramientas gráficas para crear paquetes y bases de datos.

La diferencia entre el proceso ETL y SSIS sería en que la denominación de ETL se refiere a un concepto, mientras que SSIS es la herramienta desarrollada para trabajar con ese concepto ETL.

## 4.4. SQL Server Management Studio

SQL Server Management Studio (SSMS) [18] es un entorno integrado para administrar cualquier infraestructura de SQL Server. Proporciona herramientas para configurar, supervisar y administrar instancias de SQL Server y bases de datos. Incorpora mecanismos para implementar, supervisar y actualizar los datos que usan la aplicación web desarrollada.

## 4.5. Web scraping

Como ya se ha mencionado en la descripción de las fuentes de datos, disponemos de una serie de información que podemos obtener de las políticas de privacidad en la que, sin embargo, en dos de estas la información no está accesible para poder ser descargada y manipularla como ocurren en las otras dos, a través de una API o repositorio.

Es por esto por lo que el mecanismo utilizado para obtener la información es la del *web scraping* (arañar/raspar la web) [19], donde se extraen y almacenan datos de páginas web para analizarlos o utilizarlos en otra parte. Por medio de este *scraping web* se almacenan diversos tipos de información: datos de las políticas, referencias al texto analizado y clasificado, términos de búsqueda o URLs. Estos se almacenan en bases de datos locales, tablas o ficheros en diferentes formatos.

Dentro del *scraping* hay diferentes modos de funcionamiento, diferenciándose entre el *scraping* automático y el manual. El *scraping* manual define el copiado y pegado manual de información y datos, si se desea encontrar y almacenar alguna información concreta. Es un proceso muy laborioso que raras veces se aplica a grandes cantidades de datos.

En el caso del *scraping* automático, se recurre a un software o un algoritmo que analiza diferentes páginas web para extraer información. Se utiliza software especializado según el tipo de página web y el contenido. Dentro del *scraping* automático, se diferencian varios modos de proceder:

- Analizador sintáctico: Los analizadores sintácticos, también llamados *parsers*, se utilizan para convertir un texto en una nueva estructura.

- Bots: Un bot es un software dedicado a realizar determinadas tareas y automatizarlas. Se utilizan para examinar páginas web automáticamente y recopilar datos.

- Texto: Aprovechar la función *grep* de Unix en línea de comandos para buscar en la web determinados términos en Python o Perl. Es un método sencillo para extraer datos.

En este caso, la herramienta que se ha seleccionado para hacer el *scraping* de la web una vez analizado el formato de los datos ya disponibles para OPP-115 Manual y ToS;DR es ParseHub [20]. Permite realizar el *scraping web* de cualquier sitio, indicando la URL y seleccionando los ítems interesados, navegando por distintas páginas. Las ventajas que dispone este software respecto al uso de otros serían:

- No dispone de restricción del máximo de tuplas recogidas.

- Facilidad de utilización.

- Variedad en el formato de los datos resultantes en CSV/XLS, JSON.

Realizando el *scraping* sobre la fuente de datos OPP-115 Automática, tenemos la desventaja de las interfaces en las que no hay navegación y/o recarga de las páginas. Sólo se resaltan las frases y párrafos que incluyan alguna coincidencia con lo que se está seleccionando, manteniendo la misma página. Por ello hay que definir 9 selectores, uno por cada categoría, ya que no se hace referencia en cada categoría a las mismas partes del texto, como si fuese estático.

Una vez que se realizan dos selecciones en la parte del documento en una categoría, el programa ya interpreta la selección del resto del texto que está resaltado en color. En la parte izquierda, se pueden ir nombrando las columnas de las tablas o claves del JSON y el tipo de extracción que se realiza (texto, objetos, URL, elementos HTML). Seguidamente, en la parte inferior se puede observar en todo momento la estructura resultante en tiempo real según se va realizando.

Podemos guardar la estructura de *scraping* realizado y pasar a modo navegación sin realizar ninguna selección de un texto resaltado a otro. En la figura 4.2 se puede observar con detalle la página principal sobre la cual se llevan a cabo todas estas operaciones.

Una vez que se obtiene toda la información que nos interesa, en *Get Data* el programa nos lleva al sitio de la figura 4.3 donde realiza la extracción y procesa la información. La duración depende, entre otros, del número de elementos a extraer y la conexión disponible.

Esta descripción del uso de la herramienta ParseHub para OPP-115 Automática es igual para obtener la información en la interfaz de PrivaSeer.

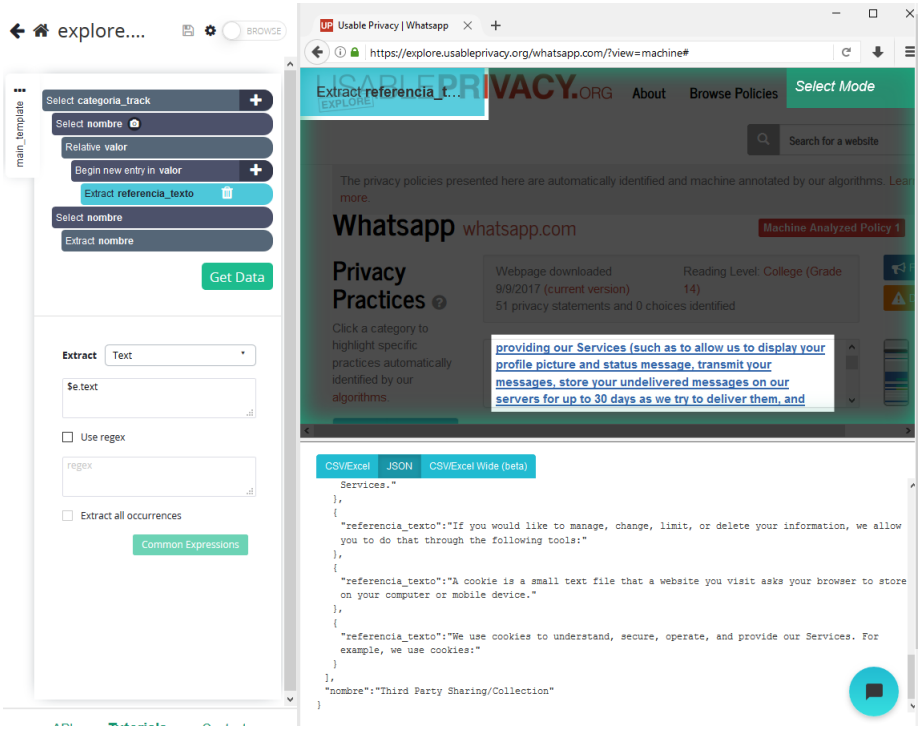


Figura 4.2: *Web scraping* con ParseHub.

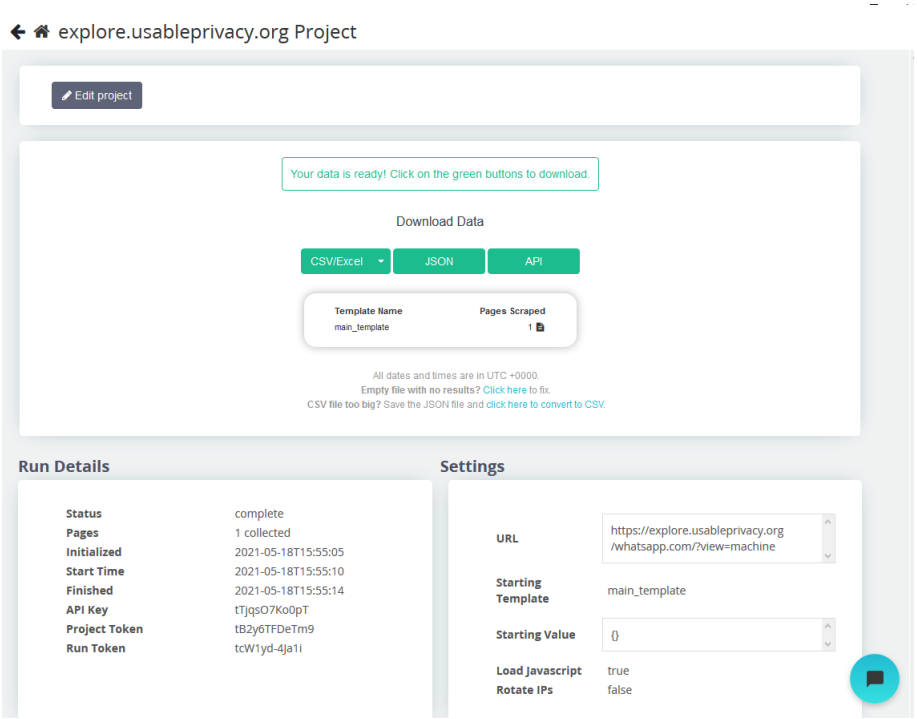


Figura 4.3: Obtención de los datos *web scraping* con ParseHub.

## 4.6. Tecnologías web

Para la construcción de una interfaz de usuario que muestre los datos al usuario, se utilizará una aplicación web. En ella, se utilizarán distintas tecnologías básicas y ampliamente extendidas en la actualidad. Al emplear una arquitectura cliente-servidor, se hace una distinción tanto para la parte del cliente como del servidor.

### 4.6.1. Tecnologías en la parte del cliente

Las tecnologías utilizadas en la parte del *Front-end* para la conversión de los datos en una interfaz de usuario con la que el usuario interactuará son:

- HTML5: La versión 5 de HTML, que nos proporciona un lenguaje de marcado de hipertexto para la creación y estructuración de secciones, párrafos, encabezados y enlaces a través de etiquetas y atributos.
- CSS3: La última versión de CSS para el control de toda la parte de estilos en los elementos escritos por medio de HTML, separando así el contenido de su representación visual. Ambas tecnologías son muy dependientes.
- JavaScript: Se utiliza la última versión 1.6 de JavaScript, junto con HTML y CSS para la implementación de funciones en las páginas web y mejorar su estructura.
- Navegador web: Conformando la parte del cliente dentro de esa arquitectura cliente-servidor, para la recuperación y representación gráfica de la Web.

### 4.6.2. Tecnologías en la parte del servidor

Las tecnologías utilizadas en la parte del *Back-end* para dar soporte al sistema que se va a crear y tener accesibilidad a los datos son:

- Apache Tomcat: La versión 9.0 de Tomcat para conseguir un contenedor de *servlets* con todas sus especificaciones y las de los JavaServer Pages (JSP), funcionando como un servidor web.
- SQL Server: La versión 15.0 de SQL Server, que nos proporciona un Sistema Gestor de Base de Datos (SGBD) relacional, para dar servicio a la aplicación a través de su almacenamiento de datos y posterior consulta.

# Capítulo 5

# Análisis

En la introducción de este proyecto ya mencionamos cual era el objetivo principal, la consulta de forma integrada de los metadatos de políticas de privacidad. Para ello, se construye un repositorio central que permita consultar información sobre cómo se realiza el seguimiento, que información se comparte con terceras organizaciones, cuáles son esas organizaciones, entre otros.

En cualquiera que sea el proyecto de software que se realice necesitamos definir unos requisitos, servicios y restricciones que tendrá el proyecto.

## 5.1. Tipos de consultas

Al almacenar la información en un almacén, debemos de alguna manera recuperar esa información, es decir, tiene que ser accesible para que la aplicación web desarrollada tenga funcionalidad. Las consultas para realizar responden a las siguientes preguntas:

- ¿Cuáles son las tecnologías utilizadas para hacer *tracking*?
- ¿Cuáles son los datos y la finalidad con la que una organización recoge la información?
- ¿Dato y propósito con el que es compartido a una tercera parte la información?
- ¿Organización tercera con la que se comparte ese dato y elecciones del usuario de rechazo o modificación de esa compartición?
- ¿Qué clasificación tiene esa política en cuanto al tratamiento de los datos y su enlace al documento?

Este tipo de preguntas nos sirve de ayuda entre la información que se ha extraído de las fuentes de datos para el Warehouse y posteriormente para describir los requisitos de información, junto con el diseño relacional de la información almacenada en la base de datos.

## 5.2. Análisis de la construcción del Warehouse

### 5.2.1. Requisitos

En las siguientes secciones, describiremos cuáles son los requisitos funcionales, no funcionales y de información que tendrán los usuarios que utilicen el sistema desde el punto de vista de la construcción de un Data Warehouse.

Para la realización de los procesos ETL, hay software especializado en la integración de la información, y se ha buscado este software (SSIS) que nos ofrece una serie de características para poder llevarlo a cabo desde el origen de los datos al destino.

#### 5.2.1.1. Requisitos funcionales

Los Requisitos Funcionales (RF) son la explicación del funcionamiento y servicio que tendrá el software utilizado en la construcción del Data Warehouse, junto con el comportamiento que tendrá, interacciones de sistemas, respuestas y procesos. Los requisitos funcionales extraídos de las fuentes de datos y la construcción del Data Warehouse a través de este software serían:

- RF-01: El sistema deberá permitir acceder a las fuentes de datos.
- RF-02: El sistema deberá permitir leer información de las fuentes de datos.
- RF-03: El sistema deberá permitir almacenar la información en el almacén.
- RF-04: El sistema deberá permitir actualizar la información del almacén.

#### 5.2.1.2. Requisitos no funcionales

Los Requisitos No Funcionales (RNF), a diferencia de los requisitos funcionales, no hacen referencia directa a funciones que presta el sistema para las características del usuario (lo que hace). Se centra en las propiedades y el funcionamiento que debe tener el sistema (cómo lo hace). Define propiedades y restricciones que debe tener el sistema de integración utilizado.

Alternativamente, definen restricciones del sistema tales como la capacidad de los dispositivos de entrada/salida y la representación de los datos utilizados en la interfaz del sistema. Tenemos tres tipos de requisitos no funcionales [21]:

- Requisitos del producto. Especifican el comportamiento del producto, como pueden ser los requisitos de desempeño en la rapidez de ejecución del sistema y cuánta memoria se requiere. Los de fiabilidad, que fijan la tasa de fallos para que el sistema sea aceptable. También están los requisitos de portabilidad y los de usabilidad.
- Requisitos organizacionales. Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador, es decir, estándares en los procesos que deben utilizarse. Requisitos de implementación como el método de diseño a utilizar.
- Requisitos externos. Se derivan de los factores externos al sistema. Incluyen los requisitos de interoperabilidad, que definen la manera en que el sistema interactúa con los otros sistemas.

Por ello, los requisitos no funcionales del software que utilizamos para la construcción de este Data Warehouse con los procesos ETL serían:

- RNF-01: El sistema deberá ser escalable en cuanto a fuentes de datos y servicios.
- RNF-02: El sistema deberá garantizar la integridad y consistencia de los datos desde las fuentes.
- RNF-03: El sistema deberá garantizar la detección de errores y recuperación de ellos.
- RNF-04: El sistema deberá implementar mecanismos de actualización.
- RNF-05: El sistema deberá garantizar un tiempo de carga de información lo más corto posible.
- RNF-06: El sistema deberá garantizar la disponibilidad en todo momento.
- RNF-07: El sistema deberá ser fácil de mantener.

#### 5.2.1.3. Requisitos de información

Tras realizar la abstracción de la información que está disponible en las fuentes de datos en capítulos anteriores y tras realizar una clasificación de la información que irá en el Warehouse, los Requisitos de Información (RI) que serán necesarios extraer desde las fuentes de datos, a través de este software utilizado para la integración serán:

RI-01: El sistema deberá almacenar información del servicio.

- Nombre
- Fuente
- Versión
- Fecha
- Clasificación

RI-02: El sistema deberá almacenar información de los datos que recopila y almacena una organización.

- Nombre
- Fuente
- Versión
- Fecha
- Dato

- Finalidad

RI-03: El sistema deberá almacenar información de las terceras partes.

- Nombre
- Fuente
- Versión
- Fecha
- Dato
- Propósito
- Tercero
- Elección

RI-04: El sistema deberá almacenar información de la forma que realiza el seguimiento.

- Nombre
- Fuente
- Categoría

## 5.3. Análisis de la aplicación de consulta

### 5.3.1. Requisitos

Una vez que hemos definido las tres categorías de requisitos que hay (funcional, no funcional y de información), y se han determinado cuáles son para obtener la información de las fuentes de datos para construir el Data Warehouse, determinamos ahora cuáles serán los requisitos que tendremos en cada una de las categorías para la construcción de la aplicación de consulta.

#### 5.3.1.1. Requisitos funcionales

- RF-01: El sistema deberá permitir mostrar el enlace a la política de privacidad.
- RF-02: El sistema deberá permitir recuperar la información de la clasificación de un servicio.
- RF-03: El sistema deberá permitir consultar por un determinado dato.
- RF-04: El sistema deberá permitir la consulta de la finalidad de un dato.
- RF-05: El sistema deberá permitir consultar por el dato compartido con terceras partes.

- RF-06: El sistema deberá permitir obtener el propósito de la información que ha sido compartida.
- RF-07: El sistema deberá permitir consultar quién es la tercera parte con la que se comparte ese dato.
- RF-08: El sistema deberá permitir recuperar información sobre la posibilidad de concesión o revocación de los datos cedidos.
- RF-09: El sistema deberá permitir consultar cómo una organización recoge la información del usuario.
- RF-10: El sistema deberá permitir la descarga/exportación de los datos buscados.

#### 5.3.1.2. Requisitos no funcionales

- RNF-01: El sistema deberá ejecutarse en cualquier navegador con HTML5.
- RNF-02: El sistema deberá desplegarse en varios Sistemas Operativos.
- RNF-03: El sistema deberá ser fácil de mantener.
- RNF-04: El sistema deberá garantizar la integridad de los datos.
- RNF-05: El sistema deberá asegurar la consistencia de la información almacenada.
- RNF-06: El sistema deberá garantizar la detección de errores y recuperación de ellos.
- RNF-07: El sistema deberá obtener información actualizada.
- RNF-08: El sistema deberá responder correctamente y comprobar las entradas incorrectas del usuario.
- RNF-09: El sistema deberá tener un tiempo de respuesta no superior a 3 segundos.
- RNF-10: El sistema deberá implementar mecanismos de seguridad de inyección de código.

#### 5.3.1.3. Requisitos de información

Los requisitos de información en este caso son los mismos que para el acceso a los datos y construcción del Warehouse ya que en ambas situaciones accedemos a la base de datos, es decir, el almacén que tiene el mismo esquema relacional.

### 5.3.2. Casos de uso

Un diagrama de casos de uso representa el comportamiento del sistema desde el punto de vista del usuario (actor). Este actor no tiene que ser una persona, puede ser cualquier proceso o sistema externo que interactúa con el sistema.

Se utiliza para mostrar la relación existente entre un actor y los requisitos funcionales descritos anteriormente, sin una descripción de las acciones que ocurren. El diagrama de casos de uso

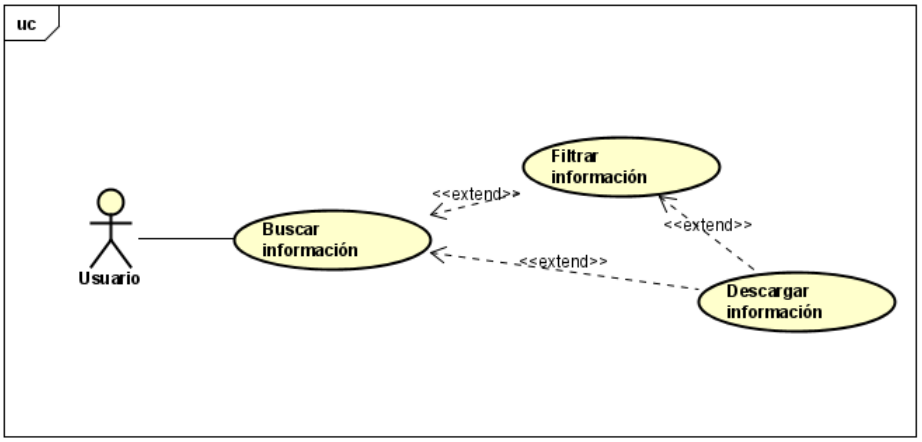


Figura 5.1: Diagrama de casos de uso.

resultante para que el usuario pueda visualizar la información con la construcción de la aplicación, sería el de la figura 5.1.

Cada uno de los casos de uso del diagrama anterior presenta una serie de acciones que determinan la forma de orientar el escenario de un caso de uso. Este escenario se divide en la secuencia normal, que define lo que se espera que ocurra o que tendría que realizarse sin ningún problema. Por otro lado, tenemos uno o varios escenarios alternativos, donde no se completa el caso de uso con éxito por alguna razón que ocurre en algún paso de la secuencia normal.

| **CU-01** | Buscar información. | |
|---|---|---|
| **Descripción** | El sistema deberá permitir al actor Usuario buscar información de un servicio. | |
| **Precondición** | El actor Usuario deberá acceder a la interfaz web por un navegador. | |
| **Secuencia** | **Paso** | **Acción** |
| | 1 | El actor Usuario selecciona la búsqueda que desea realizar e introduce los datos. |
| | 2 | El sistema comprueba los datos introducidos y muestra la información. |
| **Excepción** | **Paso** | **Acción** |
| | 2a | El sistema comprueba que no se ha introducido ningún dato, muestra un error, y el caso de uso continua en el paso 1. |
| | 2b | El sistema comprueba que se ha introducido algún dato erróneo, muestra un error, y el caso de uso continua en el paso 1. |

Tabla 5.1: Caso de uso buscar información.

| **CU-02** | Filtrar información. | |
|---|---|---|
| **Descripción** | El sistema deberá permitir al actor Usuario filtrar información de los resultados de la búsqueda de un servicio. | |
| **Precondición** | El actor Usuario deberá acceder con el navegador y tener realizada una búsqueda. | |
| **Secuencia** | **Paso** | **Acción** |
| | 1 | El actor Usuario introduce la cadena de filtro que desea realizar sobre los resultados de una búsqueda. |
| | 2 | El sistema comprueba los datos introducidos y muestra los resultados del filtro. |
| **Excepción** | **Paso** | **Acción** |
| | 2a | El sistema comprueba que se ha introducido algún dato erróneo, muestra un error, y el caso de uso continua en el paso 1. |

Tabla 5.2: Caso de uso filtrar información.

| **CU-03** | Descargar información. | |
|---|---|---|
| **Descripción** | El sistema deberá permitir al actor Usuario descargar información de la consulta de un servicio. | |
| **Precondición** | El actor Usuario deberá acceder con el navegador y tener realizada una búsqueda. | |
| **Secuencia** | **Paso** | **Acción** |
| | 1 | El actor Usuario selecciona descargar los resultados de una búsqueda. |
| | 2 | El sistema genera la descarga y lo almacena localmente la máquina del usuario, en la carpeta de descargas. |
| **Excepción** | **Paso** | **Acción** |
| | 2a | El sistema comprueba que hay problemas al generar el archivo, muestra un error, y el caso de uso queda sin efecto. |

Tabla 5.3: Caso de uso descargar información.

### 5.3.3. Modelo de dominio inicial

Un modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades [22].

De esta forma, las clases conceptuales significativas en el proyecto que estamos resolviendo sería el de la figura 5.2.

Las clases que componen el modelo de dominio de la figura 5.2 se describen como venimos haciendo en el documento como:

- Servicio: Representa la información de los servicios que estarán alojados en el sistema.
- Dato: Representa los datos de primeras partes que se obtienen de los usuarios para la política de privacidad definida de un servicio.

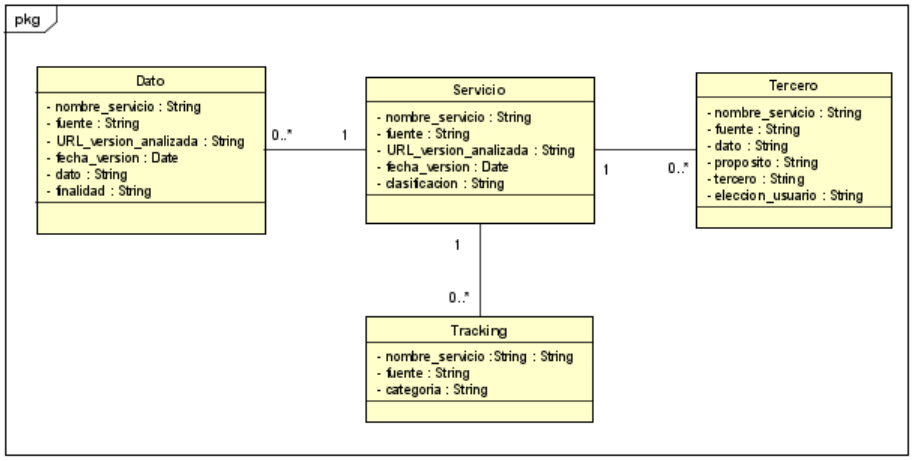


Figura 5.2: Modelo de dominio inicial.

- Tercero: Representa a los datos que son aportados en el documento de la política de privacidad, referidos a la distribución de esos datos.

- Tracking: Representa cuáles son las distintas categorías que son utilizadas por las organizaciones para hacer el rastreo de la información.

# Capítulo 6

# Diseño

En este instante ya hemos realizado el análisis conceptual, con la explicación de la funcionalidad que debe tener con los requisitos en la fase de análisis del Warehouse y la aplicación del capítulo anterior.

En este capítulo de diseño nos centraremos en la solución software para llevar a cabo la resolución del proyecto, aportando adicionalmente el conocimiento teórico.

## 6.1. Diseño de la construcción del Warehouse

### 6.1.1. Arquitectura

La arquitectura lógica nos determina cómo interactúan y se relacionan los distintos componentes del sistema.

Cuando explicamos los tipos de integración y tomamos la decisión de utilizar un Data Warehouse en capítulos anteriores, pudimos ver las diferentes fases por la que pasan los datos:

- Fuentes de datos: Es el origen de todo el proyecto. A partir de ellas se construye el sistema, que dará soporte a las consultas de información que se realicen. En PrivaSeer, ToS;DR y OPP-115 Automática, dicha información está disponible mediante ficheros en JSON, mientras que en OPP-115 Manual está presente en CSV, representados en la figura 6.1 de la arquitectura de un Data Warehouse [6].
- ETL: Sobre las fuentes de datos de las políticas de privacidad, y a través del software utilizado para la integración, se realizan los procesos ETL para la Extracción, Transformación y Carga que recogerán toda la información de los orígenes (fuentes de datos), y la carga en un nuevo repositorio central de información, el Data Warehouse, funcionando como tuberías donde la salida de un elemento es la entrada del siguiente elemento para poder obtener de él su información unificada.

### 6.1.2. Diagrama Entidad-Relación

Un diagrama de Entidad-Relación se utiliza para representar que componentes participan en el sistema y la forma en que se relacionan. El diagrama de Entidad-Relación resultante para la

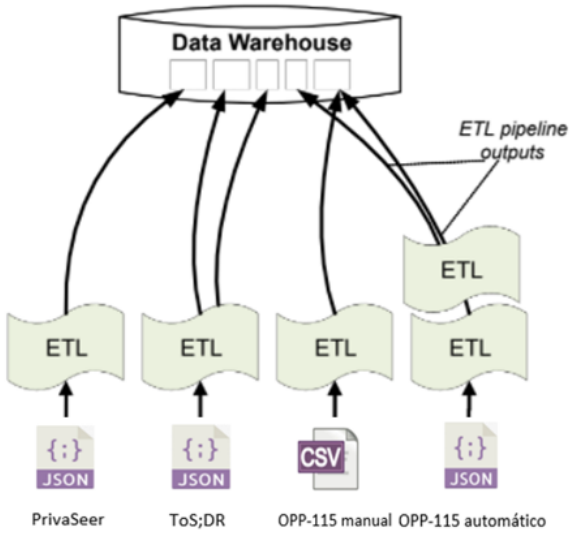


Figura 6.1: Diseño de la arquitectura lógica del Warehouse.

construcción del Data Warehouse es el de la figura 6.2. En él, está compuesto por tres elementos principales que son:

- Entidades: Tendremos una entidad por cada componente utilizado. En nuestro caso, las entidades empleadas son servicio, dato, tercero y tracking.
- Atributos: Componente fundamental del diagrama, ya que cada atributo hace referencia a las propiedades empleadas para una entidad. En la entidad servicio, los atributos empleados son: id_pol, nombre_servicio, fuente, URL_version_analizada, fecha_version y clasificacion.
- Relaciones: Vínculos que existen entre las parejas de entidades. Un servicio recoge datos, comparte con terceros y rastrea con un seguimiento.

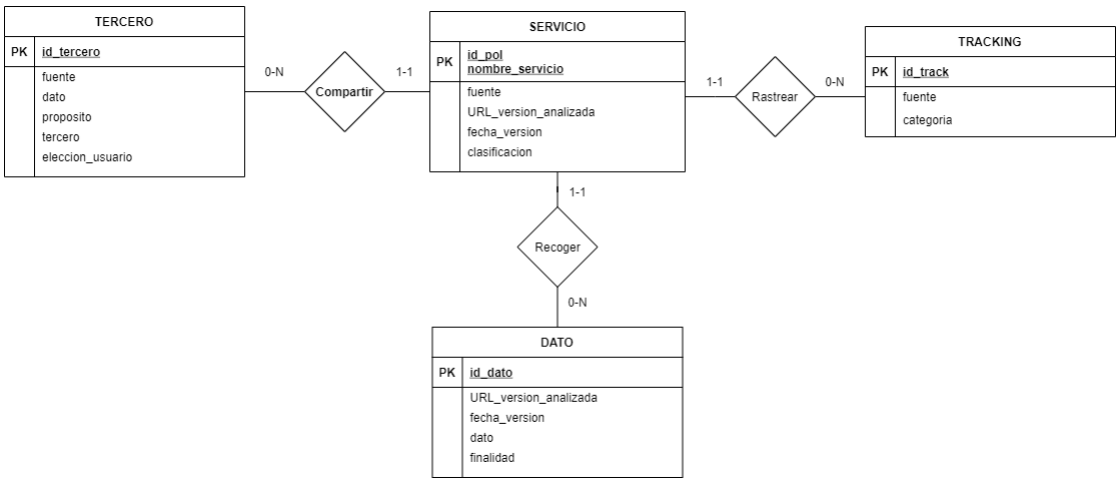


Figura 6.2: Diagrama Entidad-Relación.

El diagrama relacional resultante, mediante la transformación de uno a varios en las relaciones binarias de diagrama de Entidad-Relación de la figura 6.2, está representado en la figura 6.3. En este diagrama, el esquema relacional resultante, con la explicación de cada uno de sus atributos se describe como:

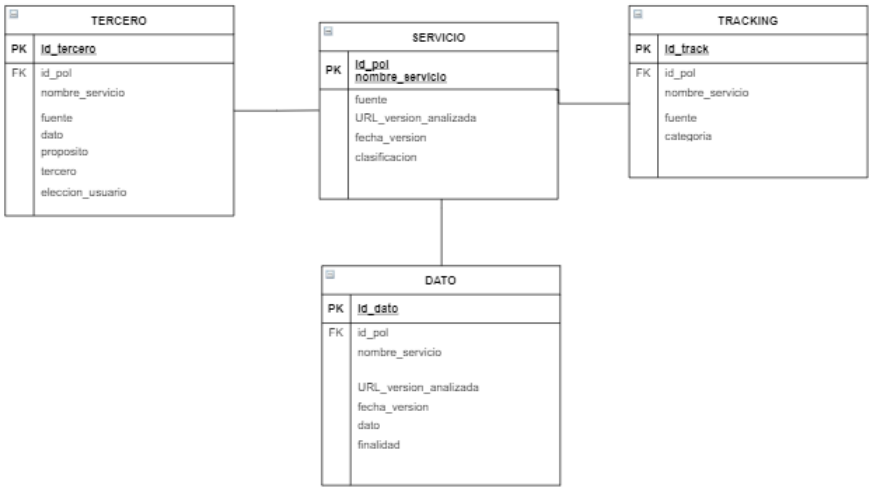


Figura 6.3: Diagrama relacional.

SERVICIO(id_pol, nombre_servicio, fuente, URL_version_analizada, fecha_version, clasificacion)

- id_pol: Identificador único que asigna el almacén a la política de privacidad que se describe.
- nombre_servicio: Nombre único del servicio de la política de privacidad.
- fuente: Enlace al documento de su política de privacidad más reciente.
- URL_version_analizada: Enlace al documento de su política de privacidad que realmente está en la base de datos.
- fecha_version: Fecha de la política cuyos metadatos están en el almacén.
- clasificacion: Valoración que le han dado los usuarios del trato justo, respeto o abuso de la información necesaria en el servicio.

DATO(id_dato, id_pol, nombre_servicio, fuente, URL_version_analizada, fecha_version, dato, finalidad)

- id_dato: Identificador único que asigna el almacén al dato recogido.
- id_pol: Identificador único que asigna el almacén a la política de privacidad que se describe.
- nombre_servicio: Nombre único del servicio de la política de privacidad.
- fuente: Enlace al documento de la política de privacidad más reciente del servicio.
- URL_version_analizada: Enlace al documento de la política de privacidad del servicio.
- fecha_version: Fecha de la política de privacidad indicada en la URL.
- dato: Nombre de dato recogido por el servicio, según se indica en la política de privacidad cuya URL aparece.

- finalidad: Texto de la política de privacidad donde se indica la finalidad para la que se recoge el dato.

TERCERO(id_tercero, id_pol, nombre_servicio, fuente, dato, proposito, tercero, eleccion_usuario)

- id_tercero: Identificador único que asigna el almacén al dato recogido y compartido con terceros.
- id_pol: Identificador único que asigna el almacén a la política de privacidad que se describe.
- nombre_servicio: Nombre único del servicio de la política de privacidad.
- fuente: Enlace al documento de la política de privacidad más reciente del servicio.
- dato: Nombre del dato recogido por el servicio, según se indica en la política de privacidad cuya URL aparece.
- proposito: Finalidad para la que se recoge el dato o que se ha encontrado en una fuente.
- tercero: Servicio al que se ceden los datos, según aparece en la política de privacidad que se describe.
- eleccion_usuario: Indica si el usuario puede optar por ceder el dato o rechazarlo.

TRACKING(id_track, id_pol, nombre_servicio, fuente, categoria)

- id_track: Identificador único que asigna el almacén al seguimiento realizado para un servicio.
- id_pol: Identificador único que asigna el almacén a la política de privacidad que se describe.
- nombre_servicio: Nombre único del servicio de la política de privacidad.
- fuente: Enlace al documento de la política de privacidad más reciente del servicio.
- categoria: Tipo de tecnología utilizada para hacer el *tracking* (seguimiento).

### 6.1.3. Diseño físico

La realización de un diseño físico nos permite optimizar el rendimiento futuro que tendrá el Data Warehouse, asegurando todas las restricciones de integridad de entidad y referencial. En esta etapa, las entidades definidas se transforman en las tablas que lo conforman, las instancias en filas y los atributos en columnas.

Como índices, las claves primarias de las tablas se transforman en índices primarios y las claves foráneas pasan a formar índices secundarios. De este modo, podremos optimizar las búsquedas. La transformación realizada, junto con los tipos de los datos utilizados, están en las tablas 6.1 a 6.4.

| **SERVICIO** | |
|---|---|
| **Atributo** | **Tipo de datos** |
| id_pol | int |
| nombre_servicio | nvarchar(50) |
| fuente | nvarchar(250) |
| URL_version_analizada | nvarchar(250) |
| fecha_version | date |
| clasificacion | nvarchar(4) |

Tabla 6.1: Diseño físico tabla Servicio

| **DATO** | |
|---|---|
| **Atributo** | **Tipo de datos** |
| id_dato | int |
| id_pol | int |
| nombre_servicio | nvarchar(50) |
| fuente | nvarchar(250) |
| URL_version_analizada | nvarchar(250) |
| fecha_version | date |
| dato | nvarchar(3000) |
| finalidad | nvarchar(3000) |

Tabla 6.2: Diseño físico tabla Dato

| **TERCERO** | |
|---|---|
| **Atributo** | **Tipo de datos** |
| id_tercero | int |
| id_pol | int |
| nombre_servicio | nvarchar(50) |
| fuente | nvarchar(250) |
| dato | nvarchar(1000) |
| proposito | nvarchar(1000) |
| tercero | nvarchar(1000) |
| eleccion_usuario | nvarchar(500) |

Tabla 6.3: Diseño físico tabla Tercero

| **TRACKING** | |
|---|---|
| **Atributo** | **Tipo de datos** |
| id_track | int |
| id_pol | int |
| nombre_servicio | nvarchar(50) |
| fuente | nvarchar(250) |
| categoria | nvarchar(500) |

Tabla 6.4: Diseño físico tabla Tracking

## 6.2. Diseño de la aplicación de consulta

En el momento de determinar cuáles eran las posibles soluciones en las que se podría aportar la funcionalidad del sistema al usuario, se consideró la utilización de Java con implementación Web. Por un lado, el usuario no tiene que descargar aplicaciones o ejecutables. Por el otro lado, Java es un lenguaje con orientación al objeto, donde podemos almacenar información que posteriormente se recuperará y mostrará al usuario.

### 6.2.1. MVC

Para definir la arquitectura software del proyecto, entre los patrones de diseño que hay y al tener una interfaz de usuario en la que el usuario interactúa con el sistema, el patrón más apropiado de utilizar es el MVC (Modelo Vista Controlador).

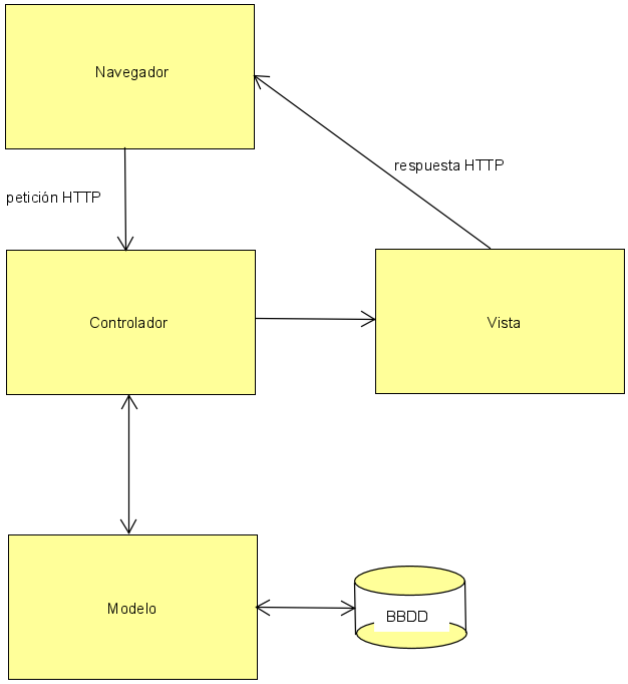


Figura 6.4: Modelo Vista Controlador.

El funcionamiento del patrón MVC de la figura 6.4 se fundamenta en:

1. El navegador del usuario (*web browser*) genera una petición HTTP.
2. El controlador, un *Servlet* que funciona a modo de servidor entre el usuario y servidor de los datos, recoge esta petición de interacción.
3. El controlador solicita los datos al modelo (las clases Java).
4. El modelo devuelve los datos tras la consulta en la base de datos.
5. El controlador selecciona una vista.

6. Se devuelve la vista que ha seleccionado al controlador.

7. El controlador proporciona la vista (JSP) al navegador con los nuevos datos cargados del modelo.

### 6.2.2. Arquitectura lógica

Para la arquitectura lógica de la figura 6.5 que determinará cómo se interactuará con el código fuente del software, se ha utilizado un diagrama de paquetes a alto nivel para su representación, sin entrar en profundidad de los distintos elementos que lo componen, cuáles son los principales eventos que recogerá el controlador (buscar información, filtrar información y descargar información) y que desencadenarán cambios en las partes más internas. Dominio con todas las clases utilizadas para almacenar la información obtenida de la base de datos. Finalmente, en Utilidades se implementará todo lo relativo con el acceso a los datos, con su conexión y la base de datos empleada.

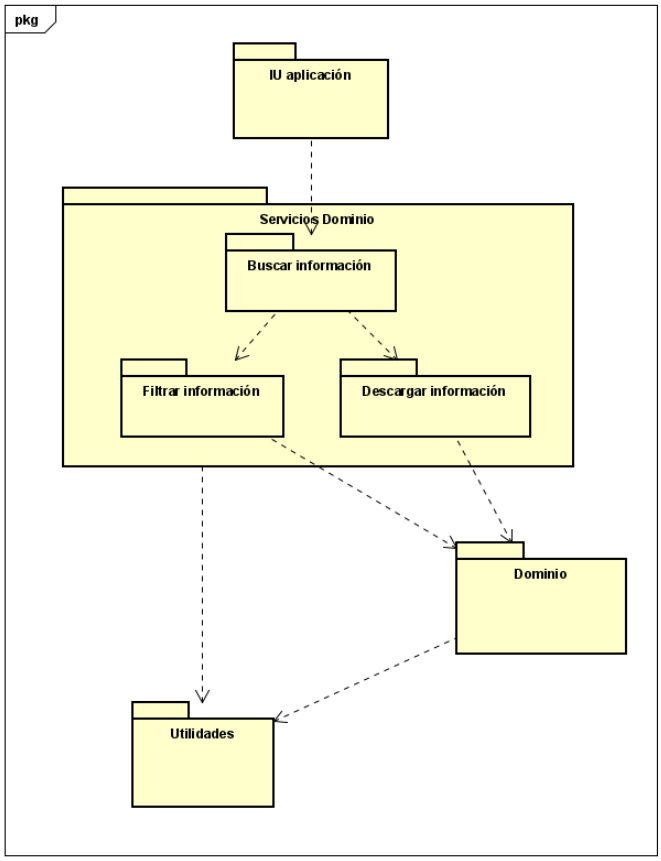


Figura 6.5: Arquitectura lógica aplicación.

### 6.2.3. Diagrama de clases de diseño

El diagrama de clases de diseño especifica las clases software que tendrá una aplicación. A diferencia de los diagramas de clases utilizados en la fase de análisis, este presenta las entidades que almacenarán información y sus conexiones con más semejanza a su implementación final. Las clases están formadas por los atributos donde almacena la información de esa entidad y las operaciones que para un determinado caso de uso es necesario llevar a cabo.

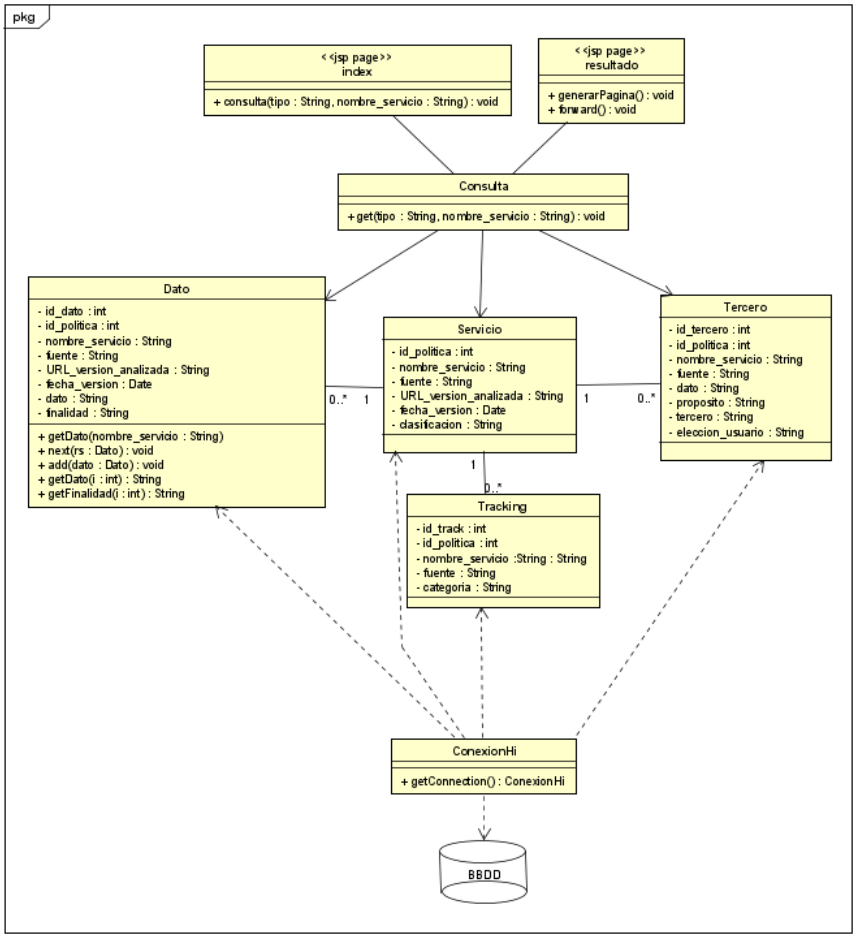


Figura 6.6: Modelo de dominio de diseño.

Este tipo de diagramas, junto con los diagramas de secuencia de la siguiente sección, implementan las operaciones que tendrán lugar a la hora de realizar los casos de uso. Para la realización de caso de uso central del proyecto con la búsqueda de información, y que se realizará dicho diagrama de secuencia en la siguiente sección, en la figura 6.6 se puede ver cómo va desde la parte más externa y visible al usuario con la que interactuará a la parte más interna. La parte externa se conforma mediante vistas, que corresponden con la página de inicio y la página de resultados obtenidos tras la consulta. La parte intermedia está compuesta por el controlador "Consulta", que revisa todas las peticiones del usuario y redirige a nuevas vistas de resultado junto con los datos. Posteriormente tenemos toda la parte más interna con el modelo, que se encarga de obtener los datos que desea el usuario en objetos solicitados por el controlador, dependiendo del tipo de consulta que realice, a través de una conexión con la base de datos. Finalmente, se encuentra la base de datos final de SQL Server, sobre la que se establecen conexiones desde el modelo para obtener resultados de las consultas del usuario.

### 6.2.4. Diagrama de secuencia

El diagrama de secuencia que se ha realizado en la figura 6.7 se corresponde con el caso de uso principal del proyecto, búsqueda de datos, más en concreto, datos de primeras partes. Este diagrama nos permite especificar el comportamiento que tendrá el sistema ante esa interacción del usuario, es decir, el desarrollo de los requisitos de la sección anterior. Permiten determinar

cuál será la secuencia de los mensajes intercambiados entre los objetos, desde el inicio hasta la finalización del caso de uso.

Cada objeto está representado como una línea de vida y focos de control, para indicar el tiempo de existencia del objeto.

El actor usuario introduce la información del servicio en la página de inicio por medio de un formulario, para consultar los datos de primeras partes. Este genera una petición HTTP de tipo GET, que es recibido por el *servlet* y a partir de aquí genera el resto de la secuencia de mensajes. Crea una lista de Dato del modelo, que obtiene una conexión de la base de datos y ejecuta la consulta para obtener los resultados de la búsqueda realizada.

Posteriormente, mientras tengamos los resultados de la consulta a la base de datos en un conjunto de resultados, se van creando objetos de tipo Dato para añadirlos a la lista de Dato. Una vez finalizado, el *servlet* selecciona la página de la vista que mostrará al usuario. Esta nueva página JSP lee los datos que recibe de la lista, y los muestra en el lugar adecuado de la página. Se reenvía la página al usuario y finalmente, tiene lugar la respuesta HTTP.

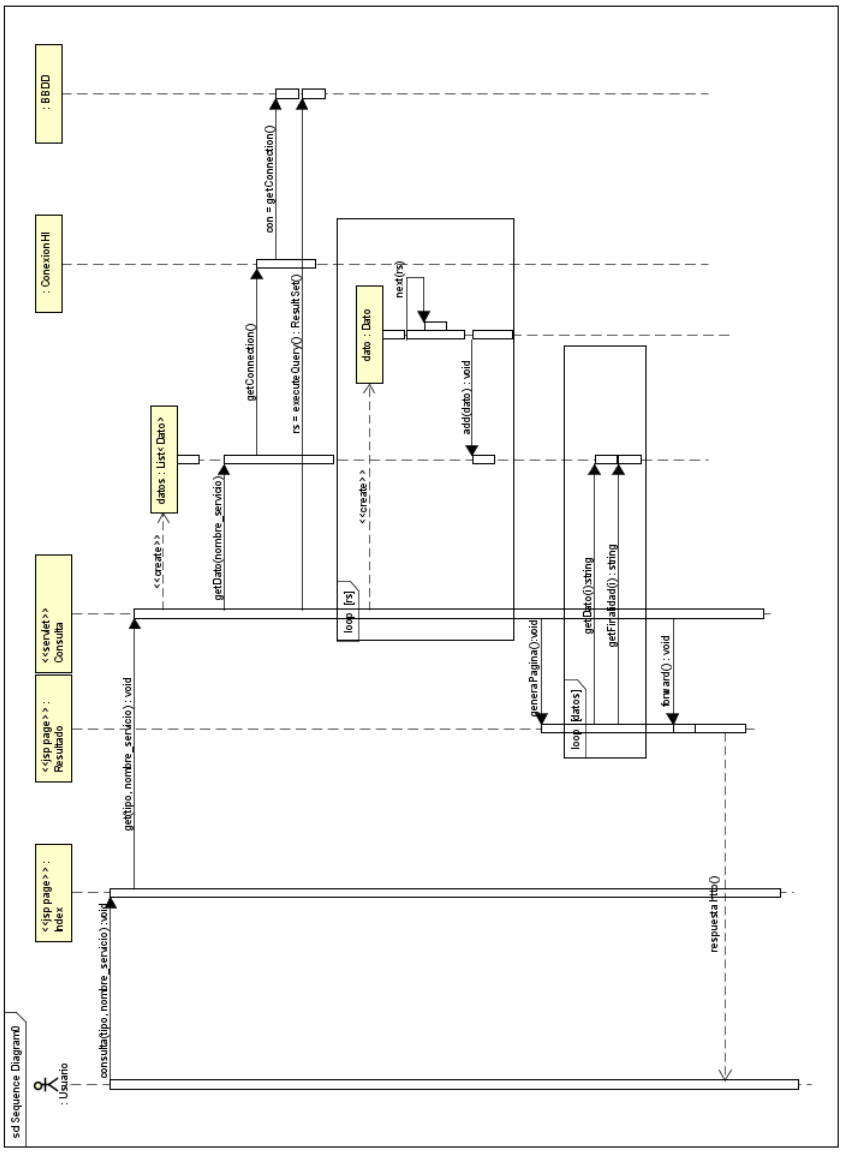


Figura 6.7: Diagrama de secuencia búsqueda de datos.

### 6.2.5. Arquitectura física

En la arquitectura física de la figura 6.8, se determina cómo se llevará a cabo su implementación en dispositivos.

El servicio web que se está ejecutando en la máquina del usuario está soportada por una base de datos donde almacena toda esta información que será actualizada en función de los datos que desea obtener. Por ello, se utiliza el esquema básico y típico sin muchas variantes de soluciones estándar para organizaciones estándar en niveles. En este caso local que estamos desarrollando, el *web browser* del cliente que tiene la interfaz de usuario, está conectado al servidor web a través de internet que a su vez se conecta al servidor de aplicaciones, para las ejecuciones del servidor de aplicaciones de la aplicación que se desarrollará con su contenido dinámico. Este servidor de aplicaciones finalmente está conectado con el servidor de base de datos para proporcionarnos la lógica dinámica usando un protocolo de conexión con la base de datos INT_BD.

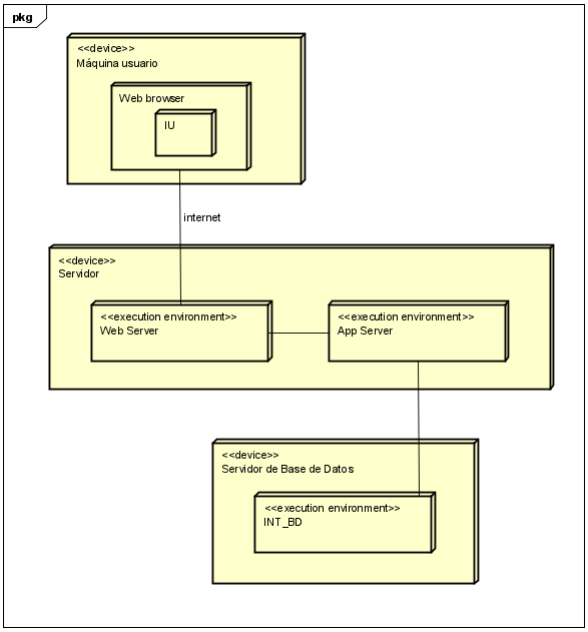


Figura 6.8: Arquitectura física aplicación.

# Capítulo 7

# Implementación

En este capítulo del proyecto se describe cómo se ha realizado el proceso final de construcción del Warehouse y la posterior aplicación web.

Como venimos haciendo en los dos últimos capítulos, el proceso está dividido en dos partes. Una la que concierne todo lo relacionado con el proceso de carga en el Data Warehouse y la otra relacionado con la puesta en funcionamiento de la consulta de la información de él.

Este capítulo es uno de los más importantes ya que se explicará todo el proceso de construcción de ambas partes, paso a paso, detallado en algunas zonas y extrapolables para el resto.

## 7.1. Implementación de la construcción del Warehouse

### 7.1.1. Creación de la estructura

Antes de empezar a realizar la carga de información en la base de datos, debemos tener construida la base de datos y las tablas que se poblarán con la información. El proceso ETL únicamente coge los datos del origen y los lleva a un destino, por lo que tenemos que definir un DDL *(Data Definition Language)*.

Este DDL define los objetos, estructuras, relaciones y restricciones que tendrán las tablas de una base de datos.

En nuestro caso, el DDL estará compuesto por cuatro consultas de tipo CREATE TABLE, una por cada una de las tablas que se van a utilizar, sus respectivas restricciones de integridad de entidad y referencial y sus índices.

### 7.1.2. Usuarios y permisos

En la etapa de análisis ya identificamos los usuarios (actores) que iban a interactuar en nuestro sistema: Usuario y Administrador. Cada uno de ellos accede a distintas partes del sistema y es necesario que tenga distinto rol en el Data Warehouse, con permisos de la tabla 7.1.

| **Usuario** | **Permisos** |
|---|---|
| Administrador | CRUD |
| Usuario | R |

Tabla 7.1: Permisos y usuarios.

Donde CRUD hacer referencia al acrónimo traducido como Crear, Leer, Actualizar y Borrar de las funciones básicas realizadas sobre una base de datos.

Desde el Sistema Gestor de Base de Datos SQL Server, se configuran los usuarios y permisos que tendrán. Para la configuración del nuevo Usuario, se siguen los siguientes pasos:

1. Accedemos como administrador al servidor final de la base de datos con SQL Server Management Studio.

2. Acedemos al directorio *Logins*, que es un subdirectorio de *Security* y seleccionamos *New Login*.

3. En el desplegable, en la pestaña *General*, introducimos los nuevos datos de inicio de sesión: nombre y contraseña.

4. En la pestaña *User Mappings* de la figura 7.1, seleccionamos la base de datos y la pertenencia al rol de la base de datos *db_datareader* y guardamos cambios.

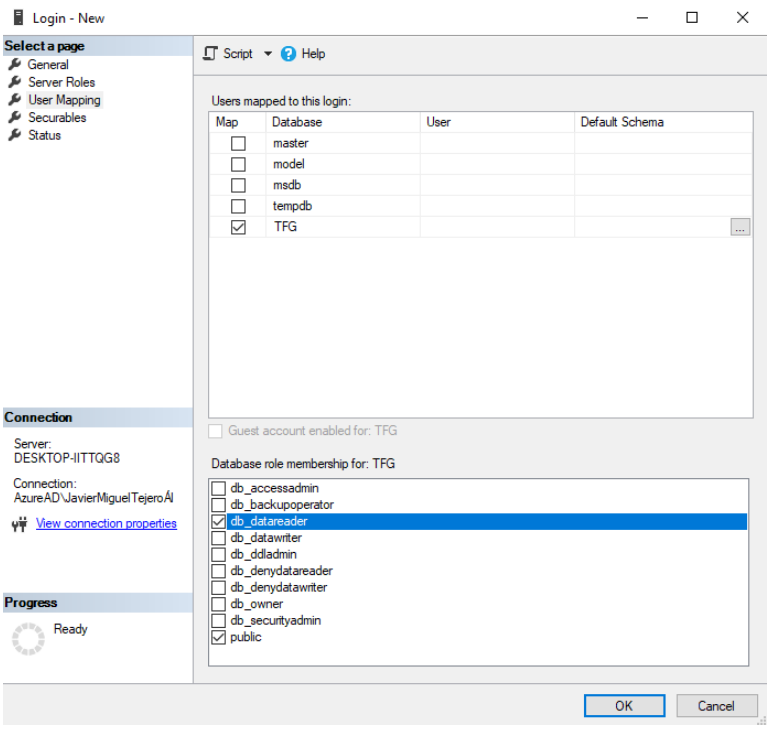


Figura 7.1: Creación de usuario y permisos.

### 7.1.3. Proceso de carga de datos

En esta sección se explican los elementos que dispone *SQL Server Integration Services* para realizar la integración de la información. Es importante diferenciar las dos estrategias de carga:

- INSERT: Si no existe un registro que ya esté insertado con mismo nombre o fuente, se inserta uno nuevo para el mismo.
- UPDATE: Si ya existe un registro insertado con un mismo nombre o fuente, se realiza una actualización de aquellos atributos que se han podido modificar.

La carga de los datos base se realiza una única vez, pero al trabajar con un Data Warehouse donde las fuentes de datos pueden realizar modificaciones, es necesario definir un mecanismo de actualización.

Para generar los procesos ETL se utiliza un software específico para la integración de información, *SQL Server Integration Services* (SSIS) que permite mover datos de origen a destino, sin modificar estas fuentes y realizar las transformaciones y cambios para incorporarlos finalmente en el destino.

Para usar esta herramienta, hacemos uso de *Visual Studio* con un proyecto de *Integration Services*. De esta manera, vamos a poder conectarnos más fácil al destino, una base de datos relacional de SQL Server.

Después de introducir los parámetros de configuración de la administración de conexiones OLEDB (SQL Server) y poder seleccionar la base de datos del servidor donde se realizará la carga, se crea un nuevo paquete de SSIS.

Una vez creado el nuevo paquete de SSIS, se incorpora al lienzo de "flujo de control" una tarea "flujo de datos". Las configuraciones que podemos añadir a este flujo son:

- **Orígenes de datos**: Los orígenes de datos se encargan de la recogida de los ficheros sobre los que se va a realizar el ETL. En nuestro caso, el tipo de ficheros será de formato JSON y CSV. El problema que tiene *Microsoft Integration Services Project* es que hay que incorporar el paquete *SSIS PowerPack* [23] para poder obtener información de diferentes formatos de ficheros que la versión inicial de MISP no dispone. Otros orígenes serán tablas de nuestro Warehouse para obtener determinados campos en la carga de ciertos datos.
- **Salida de errores**: Si se produce algún error en algún registro que no es leído, podemos redirigir esta salida de error a un fichero de texto plano. Se utilizarán en los orígenes y destinos para comprobar que la extracción y carga se realiza correctamente.
- **Conversión de datos**: El propio sistema identifica el tipo de datos que tiene el origen. Sin embargo, para hacer concordancia de tipos, podemos seleccionar manualmente aquellos atributos y el tipo de esos datos en el destino.
- **Búsqueda**: Se establece una conexión con una tabla de la base de datos de destino. Se determina el campo de comparación con el origen. Dependiendo del resultado de la comparación, podemos redirigir unas filas para un destino y otras para otro.
- **División condicional**: Nos permite redirigir el flujo de determinados registros de entrada a distintos elementos de salida, en base a condiciones.

- **Columnas derivadas**: Podremos introducir una nueva columna a la tabla temporal estableciendo esos nuevos valores.
- **Combinación**: Nos permite unir varios datos procedentes desde distintos orígenes y combinar su resultado. Similar al *JOIN* entre dos tablas.
- **Agregación**: Podemos realizar agrupación de datos por valores similares. Operaciones de recuento, GROUP BY, SUM, MAX, MIN.
- **Destino**: Una vez realizada alguna de las operaciones anteriores (transformación), es necesario cargar esos datos obtenidos en algún destino. La herramienta nos permite configurar cuáles de esos resultados se corresponden con los de la fuente destino (mapeo de las columnas de datos de destino).
- **Actualizar**: Si ya hay un registro cargado, no es necesario insertarlo de nuevo como duplicado. Será suficiente determinar si alguno de sus valores se ha visto modificado, y en su defecto actualizarlo.

### 7.1.4. Proceso ETL ToS;DR

Una vez visto todas las configuraciones con algunos de los elementos disponibles a incorporar para un flujo de datos, se procede a generarlo para cada una de las fuentes. Para la carga de SERVICIO, la representación ETL resultante desde la fuente ToS;DR sería el esquema de la figura 7.2, con su respectiva explicación de cada uno de los elementos incluidos en tres niveles. El primer nivel con la extracción. El segundo nivel y más amplio con la transformación realizada. Finalmente, el tercer nivel, con la carga final en el Warehouse.

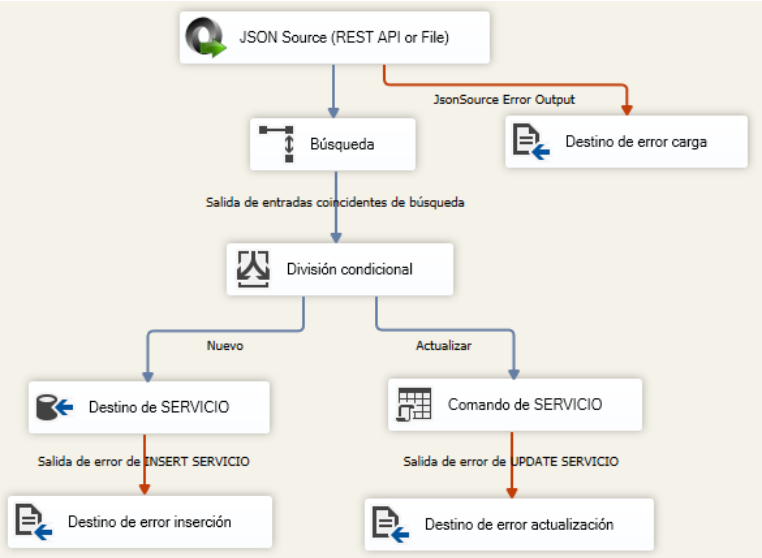


Figura 7.2: ETL ToS;DR Servicio.

1. JSON Source: Configuración del origen de la información para la extracción de un fichero en JSON de ToS;DR.
2. Búsqueda: Busca por el nombre del servicio. Similar a realizar un *JOIN* entre correspondencias de la tabla destino y datos del origen.

3. División condicional: Recibe el resultado de la búsqueda realizada y se comprueban dos condiciones. Redirige a la salida "Nuevo", aquellas filas que cumplan la condición *!ISNULL(name) && ISNULL(clasificacion)* (no ha realizado un *JOIN* en la búsqueda anterior) y redirige la salida a "Actualizar" el resto, siempre y cuando que alguno de los atributos del valor de la búsqueda (donde se ha realizado un *JOIN* por el servicio) sean distintos que el del origen (actualización). En cualquier otro caso, no se hace la actualización ni la inserción, ya que está insertado y actualizado en la base de datos.

4. Destino de SERVICIO: Es la base de datos de destino, donde recibe los registros que finalmente serán cargados en nuestra tabla SERVICIO.

5. Comando de SERVICIO: Ejecución de la consulta UPDATE, con los parámetros que recibe de la división condicional "Actualizar".

6. Destino de error carga: Fichero con información de las filas con error que no se pudieron cargar del origen y el tipo de error.

7. Destino de error inserción: Fichero con información de las filas con error que no se insertaron y el tipo de error.

8. Destino de error actualización: Fichero con información de las filas con error que tenían que actualizase y que no se pudieron actualizar y el tipo de error.

Finalmente, el mapeo de datos resultante desde las fuentes origen al Data Warehouse destino sería el de la tabla 7.2.

| **Origen** | **Destino** |
|---|---|
| $.name | nombre_servicio |
| $."links"."Privacy Policy"."url" | fuente |
| $."links"."Privacy Policy"."url" | URL_version_analizada |
| $.class | clasificacion |

Tabla 7.2: Mapeo origen-destino Servicio ToS;DR.

### 7.1.5. Proceso ETL OPP-115 Manual

Para la carga de DATO, el proceso ETL resultante sería el esquema de la figura 7.3, con su respectiva explicación de cada uno de los elementos incluidos.

La forma en que se presenta la información de esta fuente de datos (OPP-115 Manual) es particular. Por un lado, tiene una columna del fichero CSV que contiene los datos en formato JSON, cuya estructura es dinámica dependiendo de la fila analizada. SSIS únicamente analiza documentos JSON estáticos, por lo que tenemos que modularizar el acceso de esta fuente de datos. Por otro lado, el resto de las columnas de ese fichero incluye la información del servicio, como fecha del análisis, enlace. Para identificar el servicio que estamos insertando, además de tomar esta columna con los datos de ese formato, es necesario tomar el resto de las columnas para después poder filtrar por la información deseada. La explicación de los elementos utilizados para esta fuente sería:

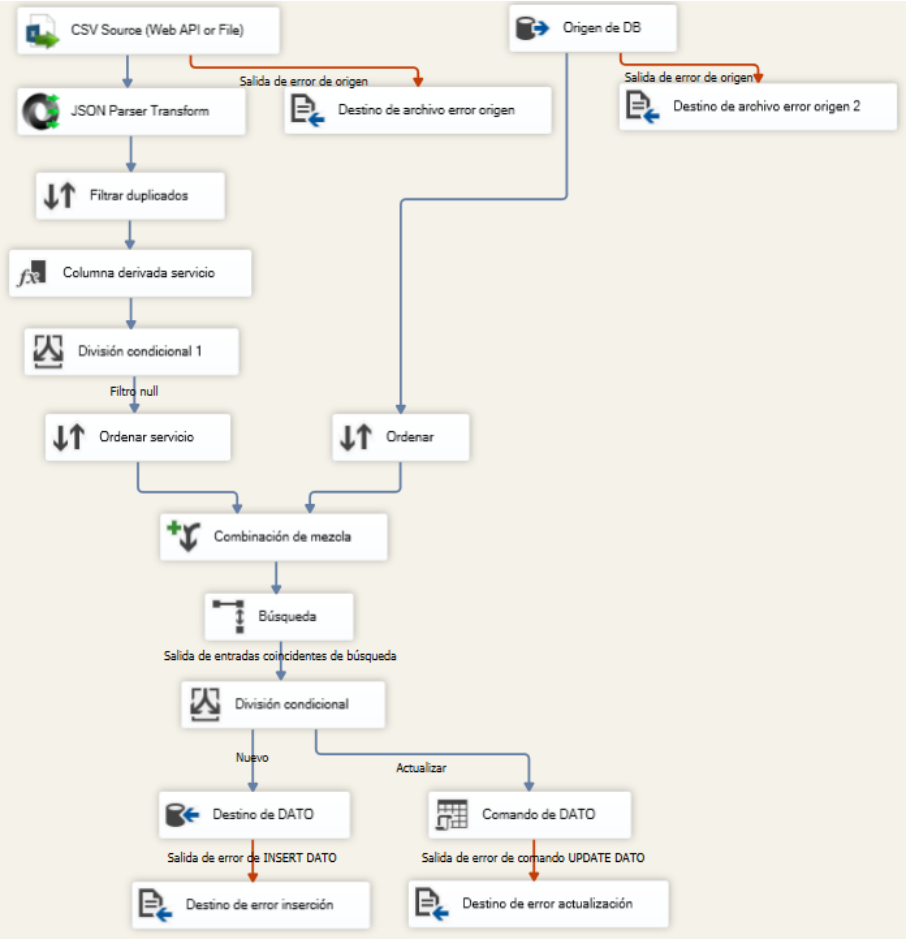


Figura 7.3: ETL OPP-115 Manual Dato.

1. CSV Source: Configuración del origen de la información para la extracción de un fichero en CSV de OPP-115 Manual.

2. Origen de DB: Datos que ya residen en el Warehouse para obtener información del servicio que se va a cargar.

3. JSON Parser Transform: Se obtiene el JSON de la columna del fichero origen CSV. Se obtienen también el resto de las columnas de fichero CSV que contienen la información de fecha del análisis, fuente y nombre del documento.

4. Filtrar duplicados: Se realizan agrupaciones donde se filtran aquellos elementos duplicados para el dato y finalidad.

5. Columna derivada servicio: Se encarga de añadir una columna de cuál es el servicio que se va a incorporar, a la tabla temporal de entrada que recibe a partir del nombre del fichero.

6. División condicional 1: Elimina los elementos de tipo *NULL* que pudiese tener del origen.

7. Ordenar: En todos los elementos de ordenación utilizados, tienen como objetivo ordenar en base a un elemento, para poder realizar una "Combinación de mezcla".

8. Combinación de mezcla: Se realiza un *JOIN* entre los elementos que se reciben tanto de la parte izquierda como de la parte derecha, ordenados, para obtener datos del servicio.

9. Búsqueda: Realiza una búsqueda con los elementos que ya están en nuestra tabla, haciendo un *JOIN* entre la tabla temporal de origen y la de destino.

10. División condicional: Recibe el resultado de la búsqueda y se comprueban que ninguno de los datos sea nulo y que el dato sea nuevo (para insertar) o existente (para actualizar). En función de su salida, redirige las filas al siguiente elemento.

11. Destino de DATO: Es la base de datos de destino, donde recibe los nuevos registros que finalmente serán cargados en nuestra tabla DATO.

12. Comando de DATO: Ejecución de la consulta UPDATE, con los parámetros de filas que se reciben de la división condicional.

13. Destino de archivo error origen: Fichero con información de las filas con error que no se pudieron cargar del origen y el tipo de error.

14. Destino de archivo error origen 2: Fichero con información de las filas con error que no se pudieron cargar de la base de datos y el tipo de error.

15. Destino de error inserción: Fichero con información de las filas con error que no se insertaron finalmente en la tabla y el tipo de error.

16. Destino de error actualización: Fichero con información de las filas con error que tenían que actualizarse y no se pudieron actualizar en la tabla y el tipo de error.

El mapeo de datos resultante desde la fuente origen al Data Warehouse destino sería el de la tabla 7.3.

| **Origen** | **Destino** |
|---|---|
| SERVICIO.id_pol | id_pol |
| SERVICIO.nombre_servicio | nombre_servicio |
| Column9 | fuente |
| Column8 | fecha_version |
| $.“Personal Information Type”.value | dato |
| $.“Purpose”.value | finalidad |

Tabla 7.3: Mapeo origen-destino Dato OPP-115 Manual.

Para la obtención de información de terceras partes desde OPP-115 Manual y cargarlo en su correspondiente tabla, la figura 7.4 muestra su proceso, similar al anterior al tener distinta estructura el JSON sobre el que se extrae una parte de la información.

1. CSV Source: Configuración del origen de la información para la extracción de un fichero en CSV de OPP-115 Manual.

2. Origen de DB: Datos que ya residen en el Warehouse para obtener información del servicio que se va a cargar.

3. JSON Parser Transform: Se obtiene el JSON de la columna del fichero origen CSV. Se obtienen también el resto de las columnas de fichero CSV que contienen la información de fecha del análisis, fuente y nombre del documento.

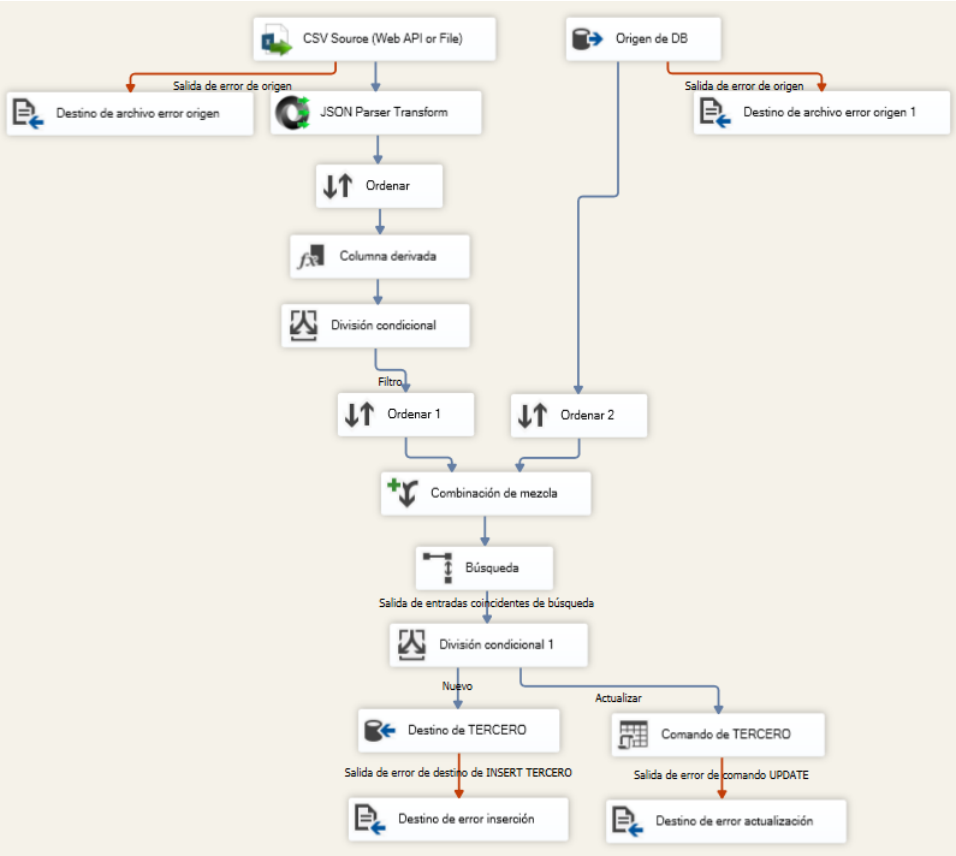


Figura 7.4: ETL OPP-115 Manual Tercero.

4. Filtrar duplicado: Se realizan agrupaciones donde se filtran aquellos elementos duplicados para el dato, propósito, terceros y la elección del usuario.

5. Columna derivada servicio: Se encarga de añadir una columna de cuál es el nombre del servicio que se va a incorporar, a la tabla temporal de entrada que recibe a partir del nombre del fichero.

6. División condicional: Elimina los elementos de tipo *NULL* que pudiese tener del origen.

7. Ordenar: En todos los elementos de ordenación utilizados, tiene como objetivo ordenar en base a un elemento, para poder realizar una "Combinación de mezcla".

8. Combinación de mezcla: Se realiza un *JOIN* entre los elementos que se reciben tanto de la parte izquierda compuesto por datos del origen como de la parte derecha que ya estaba almacenado en el Warehouse, ordenados, para obtener datos del servicio.

9. Búsqueda: Realiza una búsqueda con los elementos que ya están en nuestra tabla, haciendo un *JOIN* entre la tabla temporal de origen y la de destino.

10. División condicional 1: Recibe el resultado de la búsqueda, se comprueban que ninguno de los datos sea nulo y que el dato sea nuevo (para insertar) o existente (para actualizar). En función de su salida, redirige las filas al siguiente elemento o en cualquier otro caso, queda sin efecto.

11. Destino de TERCERO: Es la base de datos de destino, donde recibe los nuevos registros que finalmente serán cargados en nuestra tabla TERCERO.

12. Comando de TERCERO: Ejecución de la consulta UPDATE, con los parámetros de filas que recibe de la división condicional.

13. Destino de archivo error origen: Fichero con información de las filas con error que no se pudieron cargar del origen y el tipo de error.

14. Destino de archivo error origen 1: Fichero con información de las filas con error que no se pudieron cargar de la base de datos y el tipo de error.

15. Destino de error inserción: Fichero con información de las filas con error que no se insertaron en la tabla destino y tipo de error.

16. Destino de error actualización: Fichero con información de las filas con error para ser actualizadas, que no se pudieron actualizar finalmente en la tabla y el tipo de error.

El mapeo de datos resultante desde la fuente origen al Data Warehouse destino sería el de la tabla 7.4.

| **Origen** | **Destino** |
|---|---|
| SERVICIO.id_pol | id_pol |
| SERVICIO.nombre_servicio | nombre_servicio |
| Column9 | fuente |
| Column8 | fecha_version |
| $.“Personal Information Type”.selectedText | dato |
| $.“Purpose”.selectedText | proposito |
| $.“Third Party Entity”.selectedText | tercero |
| $.“Choice Type”.selectedText | elección usuario |

Tabla 7.4: Mapeo origen-destino Tercero OPP-115 Manual.

Finalmente, la última de las tablas para realizar la carga usando esta misma fuente en los atributos con la información deseada, tiene el siguiente esquema ETL de la figura 7.5.

1. CSV Source: Configuración del origen de la información para la extracción de un fichero en CSV de OPP-115 Manual.

2. Origen de DB: Datos que ya residen en el Warehouse para obtener información del servicio que se va a cargar.

3. JSON Parser Transform: Se obtiene el JSON de la columna del fichero origen CSV. Se obtienen también el resto de las columnas de fichero CSV que contienen la información de fecha del análisis, fuente y nombre del documento.

4. Filtrar duplicados: Se realizan agrupaciones donde se filtran aquellos elementos duplicados para la categoría.

5. Columna derivada servicio: Se encarga de añadir una columna de cuál es el nombre del servicio que se va a incorporar, a la tabla temporal de entrada que recibe a partir del nombre del fichero.

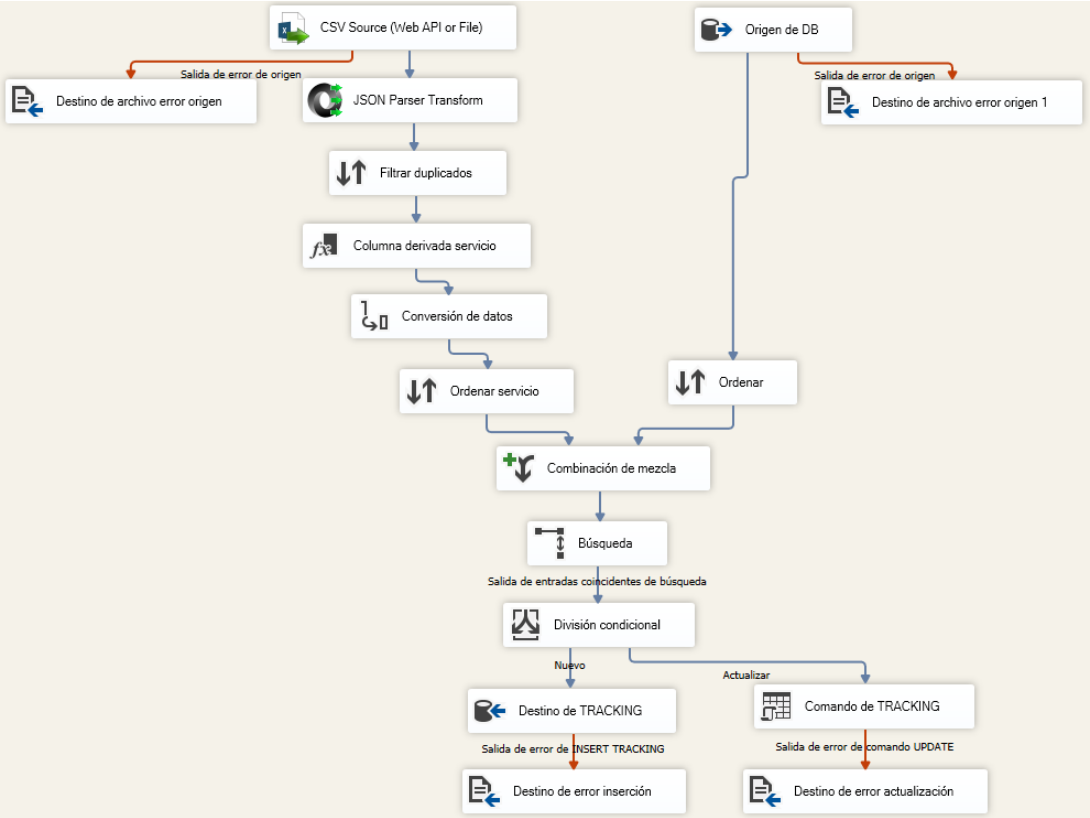


Figura 7.5: ETL OPP-115 Manual Tracking.

6. Ordenar: En todos los elementos de ordenación empleados, tiene como objetivo ordenar en base a un elemento, para poder realizar una "Combinación de mezcla".

7. Combinación de mezcla: Se realiza un *JOIN* entre los elementos que se reciben, tanto de la parte izquierda del origen de las fuentes de datos como de la parte derecha con datos ya almacenados en el Warehouse, ordenados, para obtener datos del servicio.

8. Búsqueda: Realiza una búsqueda con los elementos que ya están en nuestra tabla, haciendo un *JOIN* entre la tabla temporal del origen y la de destino.

9. División condicional: Recibe el resultado de la búsqueda, se comprueban que ninguno de los datos sea nulo y que el dato sea nuevo (para insertar) o existente (para actualizar). En función de su salida, redirige las filas al siguiente elemento.

10. Destino de TRACKING: Es la base de datos de destino, donde recibe los nuevos registros que finalmente serán cargados en nuestra tabla TRACKING.

11. Comando de TRACKING: Ejecución de la consulta UPDATE, con los parámetros de filas que recibe de la división condicional.

12. Destino de archivo error origen: Fichero con información de las filas con error que no se pudieron cargar del origen y el tipo de error.

13. Destino de archivo error origen 1: Fichero con información de las filas con error que no se pudieron cargar de la base de datos y el tipo de error.

14. Destino de error inserción: Fichero con información de las filas con error que no se insertaron en la tabla de destino y tipo de error.

15. Destino de error actualización: Fichero con información de las filas con error, que tenían que actualizarse y que no se pudieron actualizar en la tabla y el tipo de error.

El mapeo de datos resultante desde la fuente origen al Data Warehouse destino sería el de la tabla 7.5.

| **Origen** | **Destino** |
|---|---|
| SERVICIO.id_pol | id_pol |
| SERVICIO.nombre_servicio | nombre_servicio |
| Column9 | fuente |
| $.“Action First-Party”.value | categoria |

Tabla 7.5: Mapeo origen-destino Tracking OPP-115 Manual.

### 7.1.6. Proceso ETL PrivaSeer

Como ya hemos explicado en el capítulo de abstracción de las fuentes de datos, cada fuente analiza distintos servicios y granularidad de las políticas de privacidad. Por ello, se completa con información que, aunque no esté completa, permite incorporar los datos de estos servicios que han sido analizados por distinta fuente. En este caso, el ETL resultante de los datos para PrivaSeer sería el de la figura 7.6.

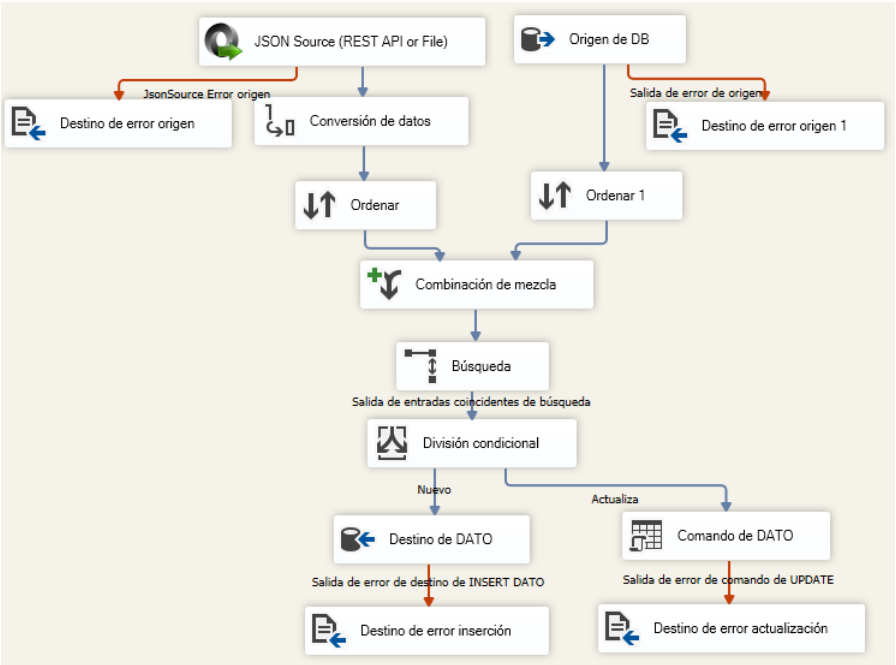


Figura 7.6: ETL PrivaSeer Dato.

La explicación de cada elemento utilizado de la figura 7.6 sería:

1. JSON Source: Configuración del origen de la información para la extracción de un fichero en JSON de PrivaSeer.

2. Origen de DB: Datos que ya residen en el Warehouse para obtener información del servicio que se va a cargar.

3. Ordenar: En todos los elementos de ordenación utilizados, tienen como objetivo ordenar en base a un elemento y eliminar los duplicados, para poder realizar una "Combinación de mezcla".

4. Combinación de mezcla: Se realiza un *JOIN* entre los elementos que se reciben tanto de la parte izquierda con el origen, como de la parte derecha con lo ya almacenado en el Warehouse, ordenados, para obtener datos del servicio.

5. Búsqueda: Realiza una búsqueda con los elementos que ya están en nuestra tabla, haciendo un *JOIN* entre la tabla temporal de origen y la de destino.

6. División condicional: Recibe el resultado de la búsqueda y se comprueban que ninguno de los datos sea nulo y que el dato sea nuevo (para insertar) o existente (para actualizar). En función la salida de su condición ("Nuevo" o "Actualiza"), redirige las filas al siguiente elemento. En cualquier otro caso (no es nuevo ni se ha actualizado), la carga queda sin efecto.

7. Destino de DATO: Es la base de datos de destino, donde recibe los nuevos registros que finalmente serán cargados en nuestra tabla DATO.

8. Comando de DATO: Ejecución de la consulta UPDATE, con los parámetros de filas que recibe de la división condicional.

9. Destino de error origen: Fichero con información de las filas con error que no se pudieron cargar del origen y el tipo de error.

10. Destino de error origen 1: Fichero con información de las filas con error que no se pudieron cargar de la base de datos y el tipo de error.

11. Destino de error inserción: Fichero con información de las filas con error que no se insertaron en la tabla destino y tipo de error.

12. Destino de error actualización: Fichero con información de las filas con error para actualizar, que no se pudieron actualizar en la tabla y el tipo de error.

El mapeo de datos resultante sería el de la tabla 7.6.

| **Origen** | **Destino** |
|---|---|
| SERVICIO.id_pol | id_pol |
| $.nombre_servicio | nombre_servicio |
| $.fuente | fuente |
| $.fecha_version | fecha_version |
| $."categoria_track" | dato |

Tabla 7.6: Mapeo origen-destino Dato PrivaSeer.

## 7.2. Implementación de la aplicación de consulta

Una vez que ya se ha realizado todo el proceso de análisis de requisitos y diseño de arquitectura lógica y física, es necesario plasmar todo en la aplicación que los usuarios podrán consultar. En este momento ya disponemos del Data Warehouse que soportará nuestra aplicación para la consulta de información.

Las librerías y los lenguajes utilizados para la construcción de la aplicación han sido:

- HML5 para la generación de vistas.
- CSS3 y JavaScript 1.6 para los diseños de la interfaz. Adicionalmente la biblioteca Bootstrap 3.3.7 para plantillas de diseño y simplificación.
- JavaScript 1.6 y jQuery 3.5.1 para todo lo relacionado con la interfaz, validaciones, interacciones y animaciones.
- HikariCP para proporcionarnos un grupo de conexiones JDBC listo para usar y reutilizar conexiones sin gastos generales.
- JSTL para utilidades de desarrollo dinámico de páginas con los JSP, consultando fácilmente los datos, pudiendo tener nuestra biblioteca de etiquetas.
- SLF4J para implementación de *logs*, una fachada que implementa los *loggers*.
- OpenCSV para la implementación de forma sencilla de documentos CSV con la descarga de los datos.

### 7.2.1. Clases Controlador

Como ya hemos visto cuando se ha realizado la descripción del patrón MVC, tendremos una serie de controladores para las acciones del usuario, que gestionan esas peticiones que se realizan. Cualquier llamada e interacción pasa obligatoriamente por un controlador.

Esta clase controlador hereda de *HttpServlet*, con dos métodos de clase importantes, el *doGet* y *doPost*. El primero recibe los parámetros por parámetro en la URL, mientras que el segundo oculta estos parámetros. En la implementación de nuestro caso, se han utilizado ambas indistintamente dependiendo de la situación, ya que la información con la que estamos tratando no es sensible.

En esta clase se produce la creación de los objetos y la posterior consulta sobre la base de datos.

Finalmente, el controlador tiene que redireccionar a una vista con la información que se ha solicitado.

### 7.2.2. Clases Modelo

Las clases modelo, como vimos en el patrón de MVC, son las encargadas de realizar las operaciones con la base de datos.

En ellas, se definen los objetos que tendrá el sistema, los atributos y los constructores con la información que se almacena en ellos. Se utilizan las clases que se han definido en los diagramas de diseño del capítulo anterior (Servicio, Dato, Tercero, Tracking).

En las consultas (*query*) que se realicen en una base de datos, se emplearán consultas parametrizadas para proporcionar seguridad y prevenir ataques de inyección de código.

### 7.2.3. Clase Conexión

Para la implementación de la conexión con el servidor de la base de datos desde la aplicación, y obtener los datos de estos en consultas, se utiliza la librería HikariCP [24]. HikariCP nos proporciona un *connection pool* (grupo de conexiones), donde permite el manejo de una colección de conexiones abiertas a una base de datos, de manera que estas conexiones puedan ser reutilizadas al realizar múltiples consultas a la base de datos.

Permite la configuración de una serie de parámetros de seguridad, como el tamaño máximo de conexiones, dónde reside el servidor con los datos, autenticación del usuario que se conecta, entre otros.

### 7.2.4. Vistas de la aplicación

Las vistas utilizadas son ficheros con extensión JSP. Utilizan CSS para la elaboración de los estilos y cómo se mostrará al usuario.

Mientras que las páginas HTML nos proporcionan la parte estática, los JSP son los que nos van a permitir crear una página web dinámica utilizando el lenguaje Java.

Con *JSP Expression Language* de JSTL, accedemos a la información de los objetos del modelo obtenidos de la base de datos, que se va a cargar en la vista. Adicionalmente se utilizan jQuery y JavaScript para funcionalidades concretas de mostrar/ocultar elementos de la vista y validación de las entradas.

# Capítulo 8

# Pruebas

Una de las etapas más importantes del proyecto una vez que ya se ha elaborado la aplicación son las pruebas. Los casos de prueba nos permiten determinar si finalmente el sistema desarrollado cumple con las especificaciones, sin errores y con el comportamiento deseado logra su cometido.

Previamente a la elaboración de la aplicación, es necesario obtener los datos con la construcción del Data Warehouse. Para su construcción, como vimos en el capítulo de implementación, se utiliza SQL Server Integration Services como software integrador. En el apéndice A se puede visualizar las pruebas que se han realizado de su correcto funcionamiento del flujo de carga y ejecución de procesos ETL.

## 8.1. Pruebas de funcionamiento general

En el lado de la aplicación, las pruebas de caja negra más significativas para el proyecto se detallan en las siguientes tablas.

Los casos de prueba de las tablas 8.1 a 8.5 se corresponden con pruebas de búsqueda de información para un servicio.

| | |
|---|---|
| **P-01** | Información y valoración de un servicio. |
| **Descripción** | Petición de información y la valoración que tiene un servicio. |
| **Entrada** | Select: Fuente y valoración del uso de la información de un servicio. Servicio: Google. |
| **Salida esperada** | Tarjeta con datos de la valoración de Google y enlace a la política de privacidad. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.1: P-01: Información y valoración de un servicio.

| **P-02** | Tecnologías de *tracking*. |
|---|---|
| **Descripción** | Petición de información de las tecnologías de *tracking* de un servicio. |
| **Entrada** | Select: Tecnologías utilizadas para hacer tracking. Servicio: Microsoft. |
| **Salida esperada** | Listado con todas las categorías de seguimiento de Microsoft. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.2: P-02: Tecnologías de *tracking*.

| **P-03** | Datos y finalidad de los datos recogidos. |
|---|---|
| **Descripción** | Solicitud de información de los datos de primeras partes y finalidades de esos datos empleados de un servicio. |
| **Entrada** | Select: Datos y finalidad por la que la organización recoge la información. Servicio: Microsoft. |
| **Salida esperada** | Acordeón desplegable con los datos y finalidad de Microsoft, clasificado por el dato recogido. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.3: P-03: Datos y finalidad de los datos recogidos.

| **P-04** | Organizaciones compartidas y revocación. |
|---|---|
| **Descripción** | Información de las organizaciones con las que se comparten datos y posibilidades de revocación. |
| **Entrada** | Select: Organizaciones con la que se comparten los datos y elecciones de usuario. Servicio: Yahoo! |
| **Salida esperada** | Acordeón desplegable con las organizaciones y elección de usuario para Yahoo!, clasificado por las organizaciones. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.4: P-04: Organizaciones compartidas y revocación.

| **P-05** | Dato compartido y propósito. |
|---|---|
| **Descripción** | Información de datos compartidos y finalidad con la que se realiza. |
| **Entrada** | Select: Dato y propósito con el que se comparte la información. Servicio: Google |
| **Salida esperada** | Acordeón desplegable con los datos y finalidad de Google, clasificado por el dato compartido. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.5: P-05: Dato compartido y propósito.

Los casos de prueba de las tablas 8.6 a 8.15 se corresponden con pruebas de filtrado de la información en base a los resultados de una búsqueda.

| | |
|---|---|
| **P-06** | Filtrar por una tecnología de seguimiento. |
| **Descripción** | Filtrar las categorías de seguimiento obtenidas para un servicio. |
| **Entrada** | Categoría: Collect. Servicio: Yahoo!. |
| **Salida esperada** | Listado con todas las categorías de seguimiento de Yahoo! que incluyen la palabra "Collect". |
| **Salida obtenida** | Salida esperada. |

Tabla 8.6: P-06: Filtrar por una tecnología de seguimiento.

| | |
|---|---|
| **P-07** | Filtrar los datos de información recogidos. |
| **Descripción** | Filtrar los datos de primeras partes que son recogidos por la organización. |
| **Entrada** | Dato: User. Servicio: Google. |
| **Salida esperada** | Acordeón desplegable de los datos recogidos por Google que contienen la palabra "User", junto con su finalidad. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.7: P-07: Filtrar los datos de información recogidos.

| | |
|---|---|
| **P-08** | Filtrar la finalidad de la información recogida. |
| **Descripción** | Filtrar la finalidad con la que los datos de primeras partes son recogidos por la organización. |
| **Entrada** | Dato: Advert. Servicio: Google. |
| **Salida esperada** | Acordeón desplegable de las finalidades de datos recogidos por Google que contienen la palabra "Advert", junto con su dato. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.8: P-08: Filtrar la finalidad de la información recogida.

| | |
|---|---|
| **P-09** | Filtrar por el dato y finalidad de la información recogida. |
| **Descripción** | Filtrar el dato y la finalidad con la que los datos de primeras partes son recogidos por la organización. |
| **Entrada** | Dato: Personal. Finalidad: Ad. Servicio: Yahoo!. |
| **Salida esperada** | Acordeón desplegable de los datos y finalidades de datos recogidos por Yahoo!, que contienen las palabras "Personal" y "Ad" respectivamente. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.9: P-09: Filtrar por el dato y finalidad de la información recogida.

| | |
|---|---|
| **P-10** | Filtrar la organización compartida. |
| **Descripción** | Filtrar las organizaciones y ámbitos con las que se comparten los datos. |
| **Entrada** | Organización: Vendors. Servicio: Microsoft. |
| **Salida esperada** | Acordeón desplegable de las organizaciones o ámbitos para Microsoft que contienen la palabra "Vendors", junto con su elección. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.10: P-10: Filtrar la organización compartida.

| | |
|---|---|
| **P-11** | Filtrar las elecciones del usuario. |
| **Descripción** | Filtrar la elección de la revocación y cesión con las organizaciones de la información. |
| **Entrada** | Elección usuario: Consent. Servicio: Microsoft. |
| **Salida esperada** | Acordeón desplegable de las organizaciones compartidas o ámbitos para Microsoft donde la elección de usuario es "Consent". |
| **Salida obtenida** | Salida esperada. |

Tabla 8.11: P-11: Filtrar las elecciones del usuario.

| | |
|---|---|
| **P-12** | Filtrar por organización y elecciones del usuario. |
| **Descripción** | Filtrar la organización y elección de la revocación y cesión con las organizaciones de la información. |
| **Entrada** | Organización: Vendors. Elección usuario: Consent. Servicio: Microsoft. |
| **Salida esperada** | Acordeón desplegable de las organizaciones o ámbitos y elecciones para Microsoft, que contienen las palabras "Vendors" y "Consent" respectivamente. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.12: P-12: Filtrar por organización y elecciones del usuario.

| | |
|---|---|
| **P-13** | Filtrar el dato compartido con terceros. |
| **Descripción** | Filtrar por los datos que son compartidos con terceras organizaciones o entidades. |
| **Entrada** | Dato: Information. Servicio: Yahoo!. |
| **Salida esperada** | Acordeón desplegable con los datos que son compartidos con terceras organizaciones por Yahoo! donde el dato filtrado es "Information", junto con su propósito. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.13: P-13: Filtrar el dato compartido con terceros.

| | |
|---|---|
| **P-14** | Filtrar el propósito del dato compartido con terceros. |
| **Descripción** | Filtrar por los propósitos de los datos que son compartidos a terceras organizaciones o entidades. |
| **Entrada** | Dato: Use. Servicio: Yahoo!. |
| **Salida esperada** | Acordeón desplegable con los datos de terceros, junto con el propósito con el que los datos que son compartidos con terceras organizaciones por Yahoo!, donde el propósito filtrado es "Use". |
| **Salida obtenida** | Salida esperada. |

Tabla 8.14: P-14: Filtrar el propósito del dato compartido con terceros.

| | |
|---|---|
| **P-15** | Filtrar el dato y propósito de la información compartida con terceros. |
| **Descripción** | Filtrar por los datos y propósitos de los datos que son compartidos a terceras organizaciones o entidades. |
| **Entrada** | Dato: Personal. Propósito: Law. Servicio: Microsoft. |
| **Salida esperada** | Acordeón desplegable con los datos de terceros, junto con el propósito con el que los datos que son compartidos con terceras organizaciones por Yahoo!, que contienen las palabras "Personal" y "Law" respectivamente. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.15: P-15 Filtrar el dato y propósito de la información compartida con terceros.

Los casos de prueba de las tablas 8.16 y 8.17 se corresponden con pruebas sobre la descarga de la información de búsqueda y filtrada.

| | |
|---|---|
| **P-16** | Funcionamiento de la descarga. |
| **Descripción** | Comprobar el funcionamiento de descarga de información. |
| **Entrada** | Select: Datos y finalidad por la que la organización recoge la información. Servicio: Microsoft. |
| **Salida esperada** | Fichero en la carpeta de descargas con los datos de la búsqueda de Microsoft en formato CSV. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.16: P-16: Funcionamiento de la descarga.

| | |
|---|---|
| **P-17** | Funcionamiento de la descarga con algún filtro. |
| **Descripción** | Comprobar el funcionamiento de descarga de información con filtros. |
| **Entrada** | Dato: Computer. Finalidad: Ad. Servicio: Google. |
| **Salida esperada** | Fichero en la carpeta de descargas con los datos de la búsqueda de Google en formato CSV. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.17: P-17: Funcionamiento de la descarga con algún filtro.

| | |
|---|---|
| **P-18** | Servicio no disponible. |
| **Descripción** | Comprobar la existencia de un servicio. |
| **Entrada** | Select: Fuente y valoración del uso de la información de un servicio. Servicio: Universidad de Valladolid. |
| **Salida esperada** | Mensaje de error que no se ha encontrado el servicio. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.18: P-18: Servicio no disponible.

| | |
|---|---|
| **P-19** | Filtro no disponible. |
| **Descripción** | Comprobar un filtro aplicado sobre los resultados de una búsqueda de dato y propósito de primeras partes. |
| **Entrada** | Servicio: Google. Dato: Valladolid. |
| **Salida esperada** | Mensaje de error que no se ha obtenido ningún resultado para ese filtro. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.19: P-19: Filtro no disponible.

| | |
|---|---|
| **P-20** | Servicio incompleto. |
| **Descripción** | Comprobar por coincidencia la existencia de un servicio. |
| **Entrada** | Select: Dato y propósito con el que se comparte la información. Servicio: Whats. |
| **Salida esperada** | Acordeón desplegable con los datos y propósitos compartidos por WhatsApp, clasificado por el dato compartido. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.20: P-20: Servicio incompleto.

| | |
|---|---|
| **P-21** | Filtro no encontrado. |
| **Descripción** | Comprobar por valores de filtro no disponibles para categorías de tracking. |
| **Entrada** | Servicio: Google. Categoría: Phone |
| **Salida esperada** | Mensaje de error que no se encontraron datos disponibles para esa búsqueda. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.21: P-21: Filtro no encontrado.

## 8.2. Pruebas de seguridad

| | |
|---|---|
| **P-22** | Funcionamiento de GET en formularios. |
| **Descripción** | Comprobar el funcionamiento del método GET por parte del usuario. |
| **Entrada** | Select: Tecnologías utilizadas para hacer tracking. Servicio: Yahoo!. |
| **Salida esperada** | En la URL debe aparecer como parámetro la información de la entrada. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.22: P-22: Funcionamiento de GET en formularios.

| | |
|---|---|
| **P-23** | Funcionamiento de POST en formularios. |
| **Descripción** | Comprobar el funcionamiento del método POST por parte del usuario. |
| **Entrada** | Categoría: Collect. |
| **Salida esperada** | En la URL no tiene que aparecer ningún parámetro con la información de la entrada. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.23: P-23: Funcionamiento de POST en formularios.

| | |
|---|---|
| **P-24** | Inyección SQL I. |
| **Descripción** | Comprobar si se detecta si el usuario intenta hacer una inyección SQL. |
| **Entrada** | Select: Dato y propósito con el que se comparte la información. Servicio: Google OR 'a'='a'. |
| **Salida esperada** | Mensaje de error indicando que se han introducido valores incorrectos. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.24: P-24: Inyección SQL I.

| | |
|---|---|
| **P-25** | Inyección SQL II. |
| **Descripción** | Comprobar si se detecta si el usuario intenta hacer una inyección SQL. |
| **Entrada** | Select: Dato y propósito con el que se comparte la información. Servicio: Google'; DROP TABLE SERVICIOS; --. |
| **Salida esperada** | Mensaje de error indicando que se han introducido valores incorrectos. |
| **Salida obtenida** | Salida esperada. |

Tabla 8.25: P-25: Inyección SQL II.

# Capítulo 9

# Conclusiones y trabajo futuro

El desarrollo del proyecto me ha servido para afianzar y poner en práctica conceptos explicados a lo largo de los estudios y adquiridos fuera de ellos. De esta manera, he realizado una labor de investigación acerca de un tema, y diferentes posibilidades de implementación en aquellos problemas que aún no se me habían planteado. Al realizar un desarrollo software completo junto con la implementación, se ha pasado por cada una de las fases del proceso de desarrollo de software.

He podido constatar una vez más, la importancia que tiene la gestión de los proyectos. Inicialmente, es necesario una buena planificación con todas las tareas y dependencias de tareas, con estimaciones del tiempo a emplear. Una identificación de los riesgos y gestionar correctamente los recursos que vamos a disponer. Posteriormente, una buena definición de los requisitos, la funcionalidad y destino final del proyecto (objetivos) nos van a ayudar a orientar el proyecto correctamente y nos han facilitado las tareas de diseño y su posterior implementación.

Se ha podido obtener conocimientos a la hora de realizar un proyecto de integración, muy importante hoy en día con la cantidad de información que tenemos disponible y en continuo crecimiento. Se han determinado en qué condiciones utilizar una determinada estrategia de integración u otra y finalmente el proceso de elaboración de un Warehouse con sus metadatos.

Cada fuente de datos es distinta y cada una almacena distinta información de los distintos servicios. No todas analizan todos los servicios de las categorías en las que nos hemos centrado (correo y mensajería) y tienen distinto nivel de grano. También se ha podido comprobar que hay un desacoplamiento entre los documentos actuales de las políticas de privacidad y cuándo se realizaron ese análisis. Sólo una de estas fuentes presenta una actualización periódica.

El software integrador utilizado para la construcción del Warehouse (SSIS), nos ha simplificado el proceso de carga de los datos en el destino, junto con una actualización futura que pueda realizarse desde las fuentes. Con las tareas de flujo realizadas y un fácil mantenimiento de esta herramienta, simplificará cualquier cambio o escalabilidad del Warehouse.

Mediante la realización de este Warehouse de políticas de privacidad, también se ha podido obtener información de la poca importancia que damos a la información que, tanto de manera consciente como inconsciente, estamos cediendo a las organizaciones sin tener conocimientos acerca de ello por no leer un documento largo y difícil de entender en servicios utilizados en el día a día.

La aplicación web está basada en una arquitectura cliente/servidor a modo de consulta, por lo que cualquier usuario sin descargas puede consultar desde el exterior la información e interactuar con la aplicación por medio de un navegador web y conexión a Internet. Sin embargo, en este

momento el despliegue es local, ya que el servidor de base de datos y de aplicación están ejecutándose en la misma máquina, accesible actualmente solo desde la misma subred. El empleo de tecnologías básicas, conocidas y estandarizadas como HTML, SQL Server y estilos hace posible su despliegue futuro. De esta manera, conseguiremos ayudar a esos usuarios con la consulta de la información integrada que puede buscar y descargar.

Como trabajo futuro y dar continuidad al proyecto elaborado, en la parte del Data Warehouse es posible ampliar las categorías de servicios sobre el cual se construye el almacén, incorporando nuevos servicios. Otra de las posibilidades es realizar un análisis manual y completar con aquella información que no ha sido posible obtener para algunos servicios de las fuentes, y que tendría que ir en el Warehouse.

Por el otro lado, en la aplicación, una de las posibilidades de implementación futura es la de dar una mayor flexibilidad a las consultas realizadas al Data Warehouse y añadir más tipos de consultas de la información.

# Bibliografía


[1] PrivaSeer. Privacy at Scale: Introducing the PrivaSeer Corpus of Web Privacy Policies. https://arxiv.org/pdf/2004.11131.pdf, 2020. [Online; accedido 24 de febrero de 2021].

[2] Parlamento Europeo y del Consejo Europeo. Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos). https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:02016R0679-20160504&from=ES, 2016. [Online; accedido 01 de marzo de 2021].

[3] California Legislative Information. Internet Privacy Requirements. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?lawCode=BPC&division=8.&title=&part=&chapter=22.&article=, 2003. [Online; accedido 01 de marzo de 2021].

[4] Anne Oeldorf-Hirsch Jonathan A. Obar. The Biggest Lie on the Internet: Ignoring the Privacy Policies and Terms of Service Policies of Social Networking Services. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2757465, 2018. [Online; accedido 01 de marzo de 2021].

[5] IBM DataStage. ¿Qué es la integración de datos? https://www.ibm.com/es-es/analytics/data-integration, 2020. [Online; accedido 16 de marzo de 2021].

[6] Z. Ives A. Doan, A. Halevy. Principles of Data Integration, 2012. [accedido 06 de marzo de 2021].

[7] Helena Ramírez. Política de privacidad web. Cumple con el RGPD en tu página. https://protecciondatos-lopd.com/empresas/politica-de-privacidad-web/, 2021. [Online; accedido 05 de abril de 2021].

[8] Tosdr. About Tosdr. https://tosdr.org/about, 2021. [Online; accedido 25 de febrero de 2021].

[9] Usable Privacy. The Creation and Analysis of a Website Privacy Policy Corpus. https://usableprivacy.org/static/files/swilson_acl_2016.pdf, 2016. [Online; accedido 25 de febrero de 2021].

[10] Usable Privacy. OPP-115 Corpus (ACL 2016). https://usableprivacy.org/static/data/OPP-115_v1_0.zip, 2016. [Online; accedido 18 de abril de 2021].

[11] William H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 3 edition, 2008. [accedido 12 de marzo de 2021].

[12] Pablo Martín Gutiérrez. Data Warehouse: Marco de calidad. https://e-archivo.uc3m.es/bitstream/handle/10016/16343/PFC_Pablo_Martin_Gutierrez.pdf?sequence=2&isAllowed=y, 2011. [Online; accedido 14 de marzo de 2021].

[13] Philippe Rigaux Marie-Christine Rousset Pierre Senellart Serge Abiteboul, Ioana Manolescu. Web Data Management. http://webdam.inria.fr/Jorge/files/wdm-data-integration.pdf, 2011. [Online; accedido 14 de marzo de 2021].

[14] Razorman. Extracción, fase clave en los procesos ETL. https://www.razorman.net/noticias-de-informatica/extraccion-fase-clave-en-los-procesos-etl.html, 2014. [Online; accedido 27 de marzo de 2021].

[15] Jerryrun. Scalable Data Warehouse Architecture. https://jerryrun.wordpress.com/2018/09/11/chapter-2-scalable-data-warehouse-architecture/, 2018. [Online; accedido 2 de Mayo de 2021].

[16] Springer. Data Warehouse Security. https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_333, 2009. [Online; accedido 02 de Mayo de 2021].

[17] Microsoft. SQL Server Integration Services. https://docs.microsoft.com/es-es/sql/integration-services/sql-server-integration-services?redirectedfrom=MSDN&view=sql-server-ver15, 2018. [Online; accedido 27 de mayo de 2021].

[18] Microsoft. Descarga de SQL Server Management Studio (SSMS). https://docs.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15, 2021. [Online; accedido 28 de mayo de 2021].

[19] Digital Guide Ionos. ¿Qué es el web scraping? https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-web-scraping/, 2020. [Online; accedido 30 de marzo de 2021].

[20] ParseHub. A free web scraper that is easy to use. https://www.parsehub.com/what_is_web_scraping. [Online; accedido 30 de marzo de 2021].

[21] Paola Caro. Técnicas para identificar Requisitos Funcionales y No Funcionales. https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales, 2012. [Online; accedido 02 de Mayo de 2021].

[22] Craig Larman. *UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* Pearson, 2 edition, 2003. [accedido 02 de mayo de 2021].

[23] zappysys. SSIS Powerpack. https://zappysys.com/products/ssis-powerpack/, 2021. [Online; accedido 27 de marzo de 2021].

[24] brettwooldridge. HikariCP. https://github.com/brettwooldridge/HikariCP, 2021. [Online; accedido 01 de Mayo de 2021].

[25] Microsoft. Agente SQL Server. https://docs.microsoft.com/es-es/sql/ssms/agent/sql-server-agent?view=sql-server-ver15, 2021. [Online; accedido 16 de junio de 2021].

# Apéndice A

# Pruebas para el proceso de carga

Este apéndice muestra las pruebas realizadas sobre el software de integración de SQL Server Integration Services, y podemos visualizar su correcto funcionamiento de la implementación de los procesos ETL.

El proceso ETL totalmente desarrollado con todo el flujo de datos disponible sería el representado en la figura A.1, donde podemos visualizar que se ha ejecutado correctamente y sin ningún error.


Figura A.1: Pruebas ETL flujo de datos.

No obstante, entrando más en detalle en alguna de las tareas de este flujo de datos (figuras A.2 a A.4), podemos ver el número de filas del que se parte en el origen en la extracción y finalmente van pasando por la transformación, filtrándose hasta la carga. En los correspondientes ficheros de salida de error, como ya se indica en las figuras, están vacíos. La figura A.5 nos permite comprobar la actualización de la información de la clasificación de ese servicio.

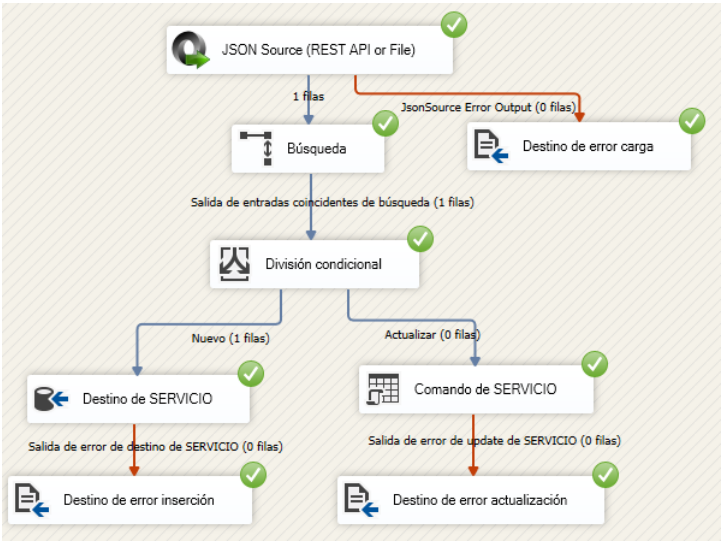


Figura A.2: Pruebas ETL carga Servicio ToS;DR.

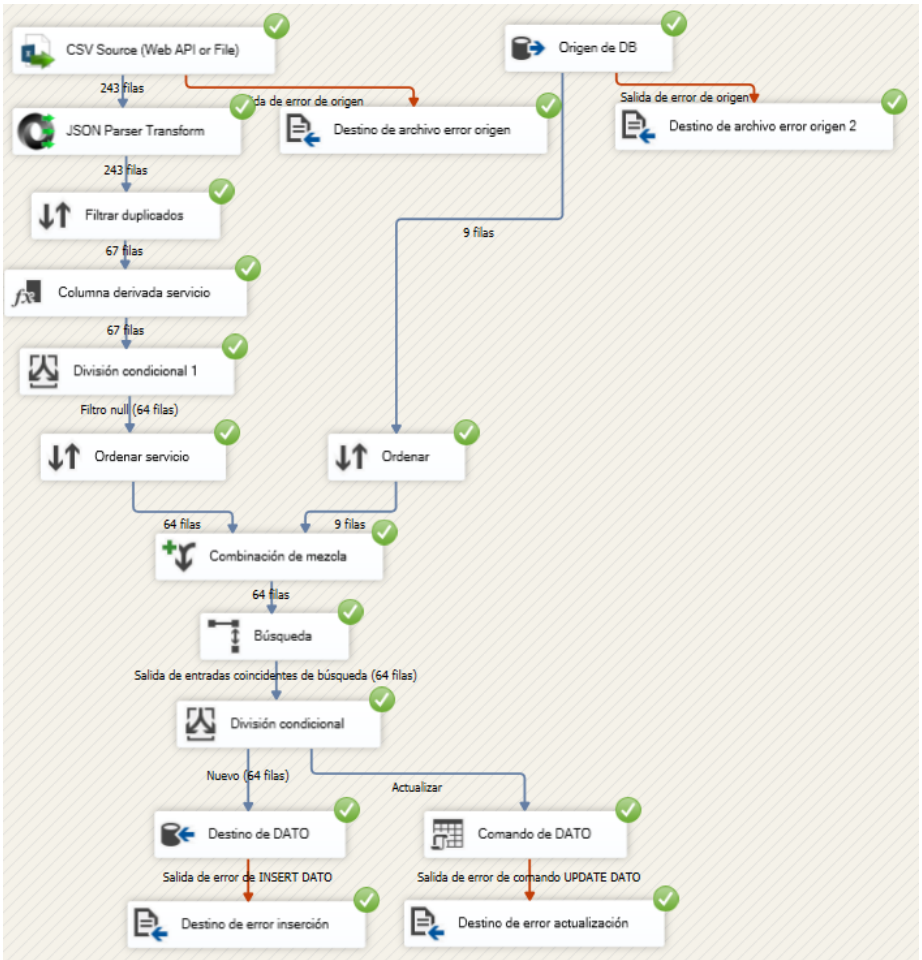


Figura A.3: Pruebas ETL carga Dato OPP-115.

Las pruebas aquí consistirían en comprobar que el proceso se ha realizado correctamente en cada tarea de flujo, no hay errores en los ficheros de salida de error, y que los registros que tiene el fichero origen se corresponden con lo que se ha introducido en la base de datos, ejecutando consultas en el sistema gestor e inspeccionando los orígenes para ver si se corresponden los valores.

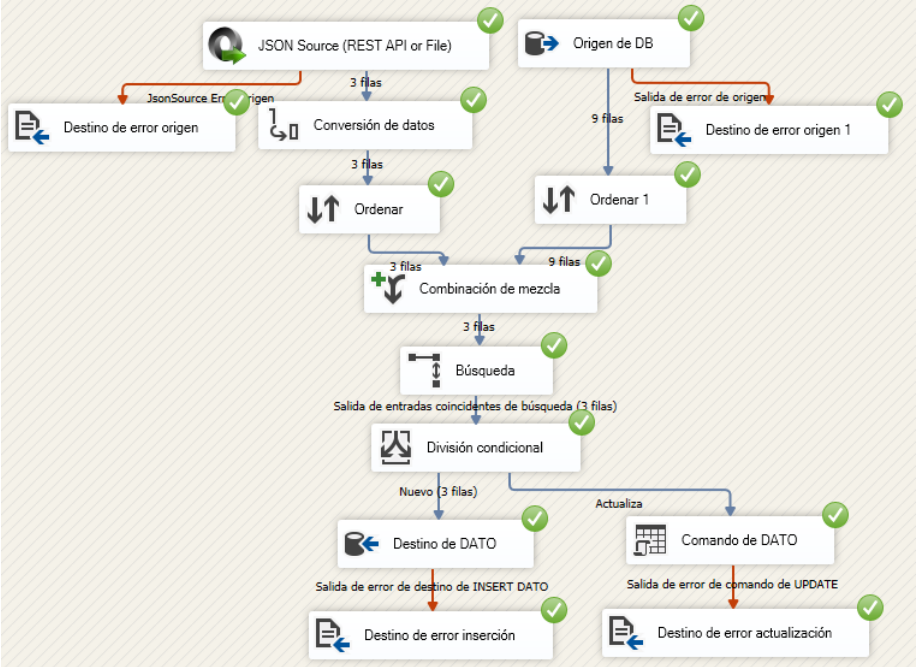


Figura A.4: Pruebas ETL carga Dato PrivaSeer.

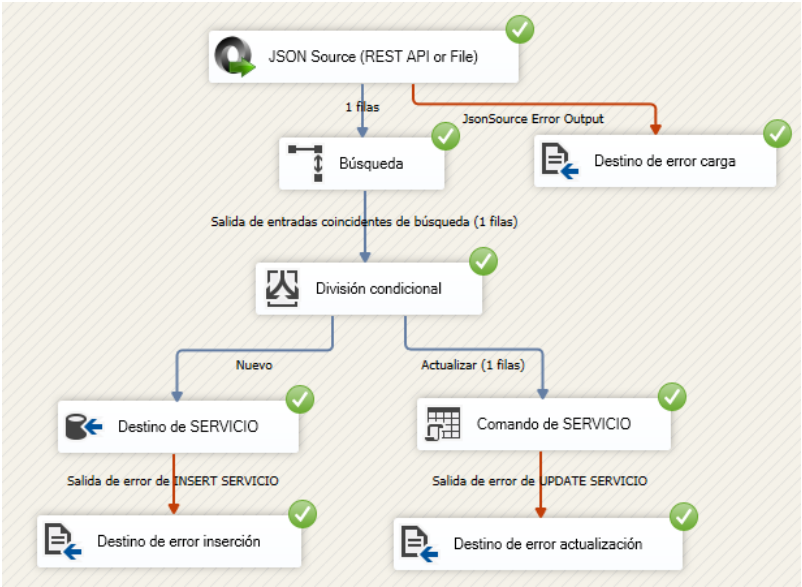


Figura A.5: Pruebas ETL actualización Servicio ToS;DR.

# Apéndice B

# Copia de seguridad y actualización

Este apéndice describe todo lo relacionado con la seguridad física mediante copias de seguridad y actualización del Data Warehouse. Una de las funcionalidades que dispone el gestor SQL Server Management Studio (SSMS) es el Agente SQL Server [25], que es un servicio de Microsoft que ejecuta tareas administrativas programadas, denominadas trabajos en SQL Server. Con este agente conseguimos planificar tareas que se realicen cada cierto periodo de tiempo.

## B.1. Copia de seguridad

En nuestro caso, podemos programar la realización automática de copias de seguridad del Data Warehouse creado, independientemente si el administrador lo realiza en un determinado instante. El proceso de copia de seguridad consiste en:

1. Iniciar el Agente de SQL Server.

2. En el directorio trabajos, crear un nuevo trabajo.

3. En *Step*, definimos un nuevo paso con el script *Transact-SQL* (T-SQL) de la copia de seguridad a realizar sobre la base de datos.

4. En *Schedules*, se programa la tarea de pasos definida en el punto anterior para que se ejecute periódicamente.

Las figuras B.1 y B.2 representan los pasos 3 y 4 anterior, respectivamente. En la primera figura, se muestra la vista general para la configuración del *step* con un T-SQL del *backup* de la base de datos utilizada como Warehouse. En la segunda figura, tiene lugar la configuración de toda la programación, donde esta tarea programada es: periódica, los lunes cada 2 semanas al mediodía y sin fecha de finalización de esta tarea programada.

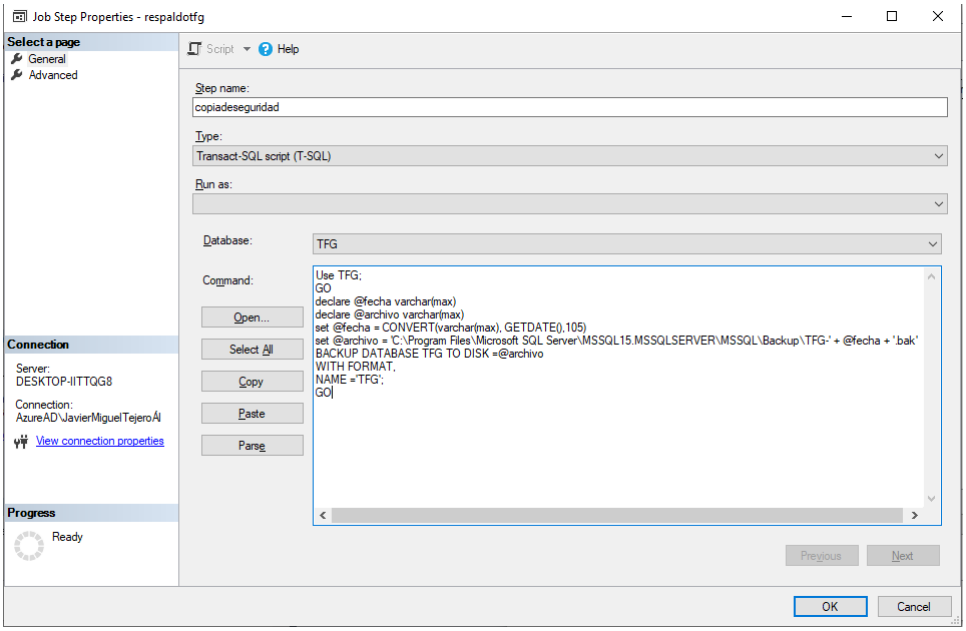


Figura B.1: Tarea copia de seguridad del agente en SSMS.

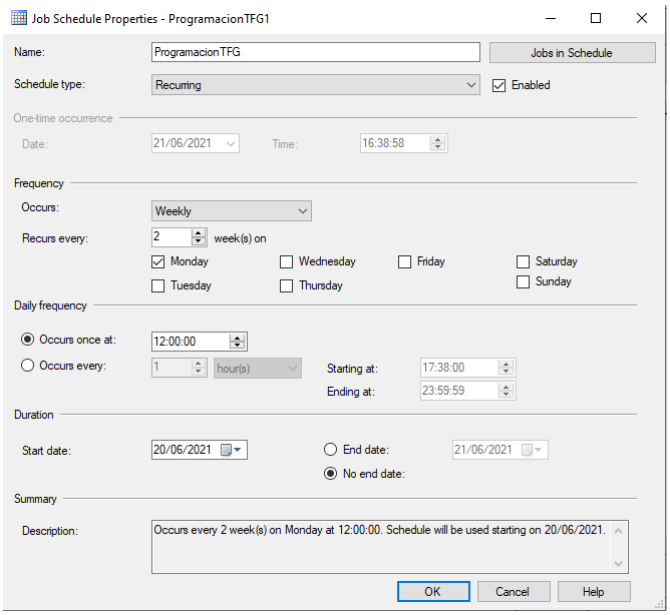


Figura B.2: Programación de la tarea copia de seguridad del agente en SSMS.

## B.2. Actualización

Es posible programar automáticamente el proceso de actualización del Data Warehouse (ETL), independientemente si lo hacemos nosotros como administrador en un determinado instante. Después de haber cargado el paquete con todos los flujos de datos de los procesos ETL en SSMS mediante la creación de un catálogo de SSIS, los nuevos pasos a seguir son:

1. En el directorio trabajos, crear un nuevo trabajo.

2. En *Step*, definimos un nuevo paso para que se ejecute automáticamente por el agente. En la figura B.3, el tipo de paquete es de SQL Server Integration Services y se ejecuta con el agente de servicio de SQL Server. El paquete es SSIS Catalog que se ha creado el catálogo de SSISDB, al realizar la carga del paquete con todos los componentes del proceso ETL.

3. En *Schedules*, se programa la tarea de pasos definida en el punto anterior siguiendo la política de actualización definida. Esta tarea programada es: periódica, los lunes cada 2 semanas antes del mediodía (antes de una copia de seguridad) y sin fecha de finalización de esta tarea programada.

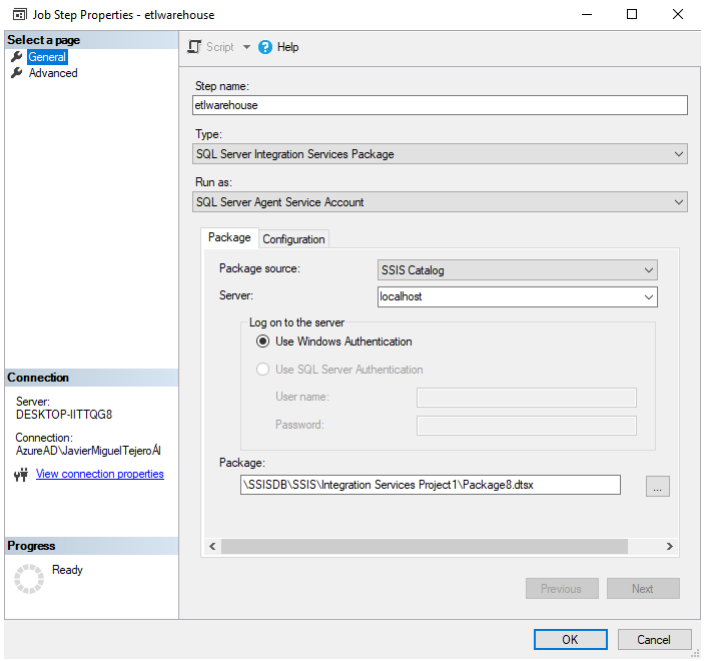


Figura B.3: Tarea actualización del agente en SSMS.

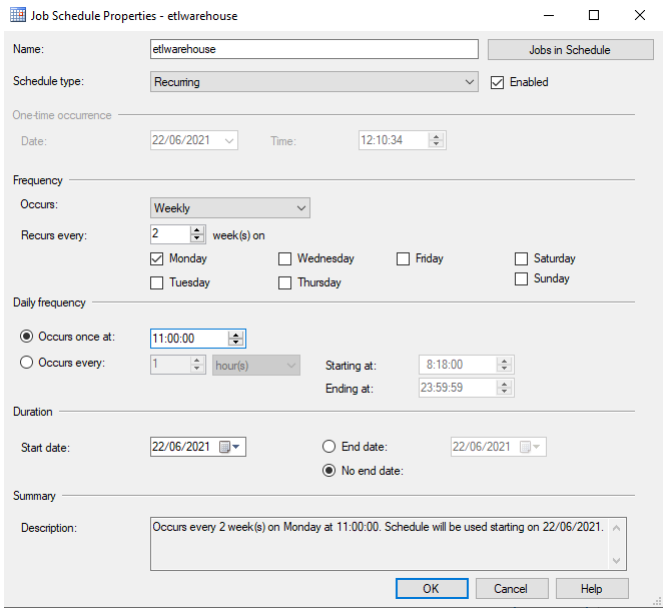


Figura B.4: Programación de la tarea actualización del agente en SSMS.

# Apéndice C

# Manual de usuario

En este apéndice se presenta el manual de usuario, donde se explica la funcionalidad que se ha implementado en el sistema, es decir, como realizar determinadas tareas.


Figura C.1: Página de inicio de la búsqueda.

En la figura C.1 podemos visualizar la pantalla de inicio principal, donde un usuario puede introducir la búsqueda que desea realizar. En ella, se comprueba si el servicio ha sido analizado y está disponible o si se ha introducido algo erróneo no válido.

Introduciendo los datos por los que se desea buscar, en nuestro caso obtener información de los datos y finalidad de primeras partes por la que la organización recoge la información para Microsoft y pulsando en buscar, nos aparece un acordeón desplegable de la figura C.2 clasificado por el dato y su correspondiente finalidad. También la fecha de incorporación y el documento de donde se ha recogido esa información.

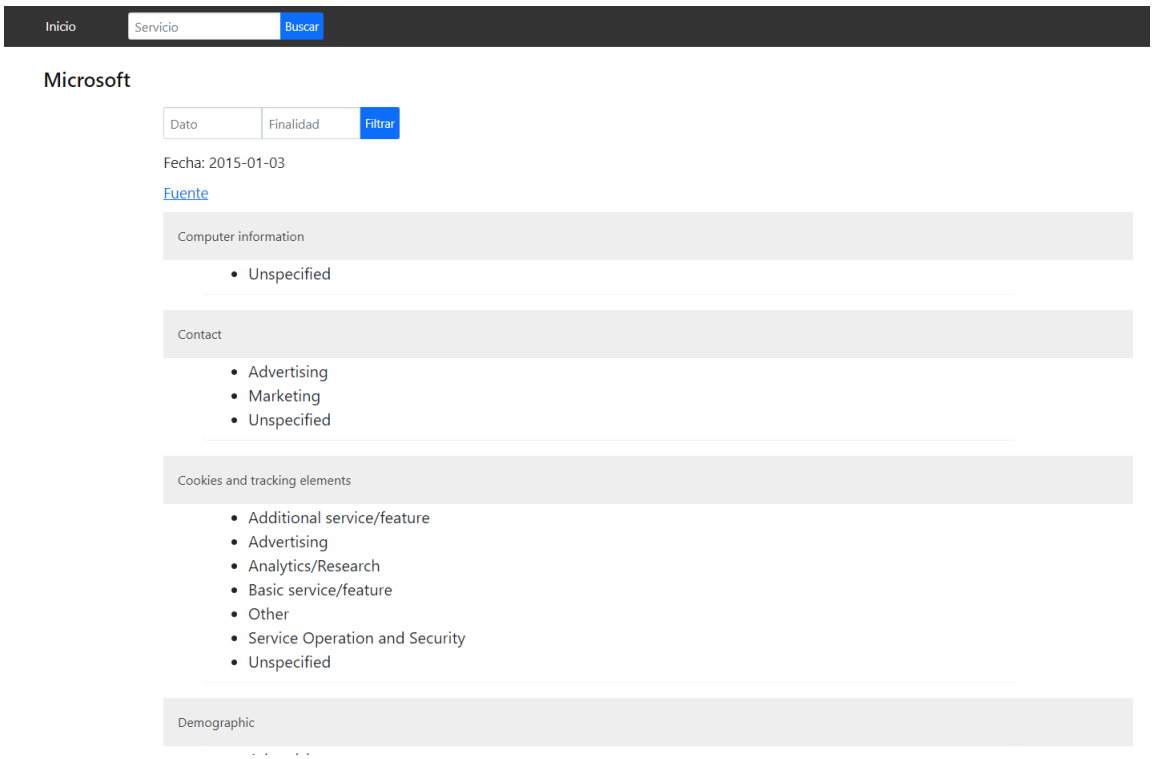


Figura C.2: Resultados de la búsqueda.

Por otro lado, si en algún momento queremos información mucho más específica y detallada de la que nos presenta, podemos realizar una búsqueda con filtro como en la figura C.3 y obtener mayor grado de detalle. En este caso, el filtro aplicado ha sido de *cookies* y *ad* para datos y finalidad, respectivamente.

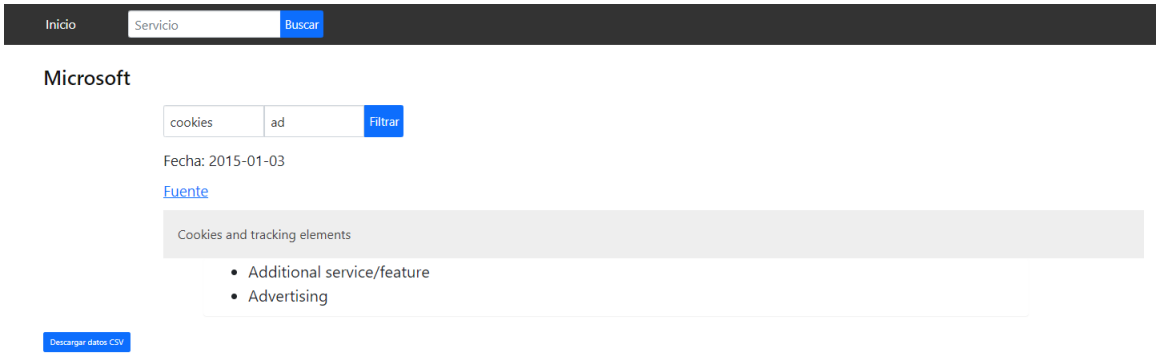


Figura C.3: Aplicación de filtro a la búsqueda.

Siempre va a tener la posibilidad de comprimir/descomprimir el acordeón de una búsqueda por los términos clasificados, volver al inicio para realizar una nueva búsqueda o eliminar cualquiera de los filtros aplicados pulsado sobre el nombre del servicio que ha buscado.

Sea cual sea la búsqueda o filtro aplicado que se realice sobre los resultados de una búsqueda, siempre vamos a tener la opción de exportar esos datos de ese servicio mediante la descarga de esos datos en formato CSV para una posterior manipulación del usuario, almacenándolo localmente en su carpeta de descargas con una estructura de la figura C.4.

```
"nombre_servicio","fuente","URL_version_analizada","fecha_version","dato","finalidad"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Computer information","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Contact","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Contact","Marketing"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Contact","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Analytics/Research"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Basic service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Other"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Service Operation and Security"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Cookies and tracking elements","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Demographic","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Analytics/Research"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Basic service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Legal requirement"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Personalization/Customization"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Service Operation and Security"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Generic personal information","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","IP address and device IDs","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","IP address and device IDs","Basic service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","IP address and device IDs","Other"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","IP address and device IDs","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Location","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Location","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Location","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Other","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Other","Other"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Analytics/Research"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Basic service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Legal requirement"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Other"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Personalization/Customization"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Service operation and security"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","Unspecified","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Advertising"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Analytics/Research"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Basic service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Other"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Personalization/Customization"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User online activities","Unspecified"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User profile","Additional service/feature"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User profile","Personalization/Customization"
"Microsoft","https://privacy.microsoft.com/en-us/privacystatement","http://www.microsoft.com/privacystatement/en-us/BingandMSN/default.aspx","2015-01-03","User profile","Unspecified"
```

Figura C.4: Estructura de descarga de la información en CSV.

# Apéndice D

# Documentación adicional

En la entrega de ficheros adicionales, se incluyen:

- **Memoria:** Versión en formato PDF de la memoria del TFG.
- **Proceso ETL:** Fichero del paquete de SSIS con todo el proceso de implementación del ETL desde las fuentes al destino.
- **Ficheros fuente de datos:** Directorio para cada fuente de datos con los ficheros de datos utilizados en la integración.
- **Script de la base de datos:** Incluye el DDL y DML obtenido después de aplicar el proceso ETL.
- **Código fuente:** Incluye todos los ficheros con la funcionalidad de la parte de la aplicación.